HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences

Tapani Hyvämäki

TESTING BAYESIAN NETWORKS AND DENSITY BASED CLUSTERING IN MAINTENANCE FAULT DETECTION

Master´s thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics.

Espoo, April 29, 2009

Supervisor: Professor Raimo P. Hämäläinen
Instructor: Licentiate of Science in Technology Vesa Hölttä

Author: Tapani Hyvämäki
Faculty: Faculty of Information and Natural Sciences
Degree Programme: Engineering Physics
Major Subject: Systems and Operations Research
Minor Subject: Applied Physics

Title: TESTING BAYESIAN NETWORKS AND DENSITY BASED CLUSTERING IN MAINTENANCE FAULT DETECTION
Title in Finnish: BAYESILAISTEN MENETELMIEN JA KLUSTEROINTI-MENETELMIEN TESTAUS KUNNOSSAPIDON VIANETSINNÄSSÄ
Chair: Mat-2 Applied Mathematics

Supervisor: Professor Raimo P. Hämäläinen
Instructor: Licentiate of Science (Technology) Vesa Hölttä

Data-driven condition monitoring of cut-to-length forest harvesters has developed to a state where substantial amounts of high quality data are available from the harvesting process and especially from the harvester head, which is the main functional part of the harvester. However, the methods that are capable of extracting the essential information from the data are relatively immature. Methods from the field of industrial process monitoring have been applied to the forest harvesting process, but so far with little success. The problem with these methods is that the variation in environmental conditions and the contribution of the human operator have a great influence on both process performance and efficiency. To date, the development of means for measuring these factors has not reached a desired level.

This thesis introduces three previously unapplied methods for data-driven condition monitoring on the forest harvester head performance index data. These methods have been used in the process industry earlier. One of the introduced methods is a density based clustering method and the other two are probabilistic methods called the Gaussian mixture and the Bayesian network models. The starting point of the analysis involves determining the distribution of the data, finding patterns in the data and identifying dependencies between the index variables. Further, based on these observations the process in-control and out-of-control states, including the fault states and the related variables, are explored.

The theoretical part of this thesis introduces forest harvester operation and the collected data, basic concepts of data-driven condition monitoring as well as the data-driven condition monitoring methods and related multivariate statistics. The experimental part applies the introduced condition monitoring models to the index data followed by an analysis of the models' suitability. The final conclusions present the findings that contain qualitative observations and recommendations about the models and the data. The main result is that the data is not sufficient to be used with the condition monitoring methods examined in this thesis. Finally, the main findings are listed and recommendations for overcoming the shortcomings are proposed. These results can be utilized in the future research of maintenance fault detection of forest harvesters.

Number of Pages: 75
Keywords: forest harvester, data analysis, condition monitoring, density based clustering, Gaussian mixture model, Bayesian network

Department fills
Approved: **Library Location:**

Tekijä: Tapani Hyvämäki
Tiedekunta: Informaatio- ja luonnontieteiden tiedekunta
Koulutusohjelma: Teknillinen fysiikka
Pääaine: Systeemi- ja operaatiotutkimus
Sivuaine: Sovellettu fysiikka

Työn nimi: BAYESILAISTEN MENETELMIEN JA KLUSTEROINTI-MENETELMIEN TESTAUS KUNNOSSAPIDON VIANETSINNÄSSÄ
Title in English: TESTING BAYESIAN NETWORKS AND DENSITY BASED CLUSTERING IN MAINTENANCE FAULT DETECTION
Professuuri: Mat-2 Sovellettu Matematiikka

Työn valvoja: Professori Raimo P. Hämäläinen
Työn ohjaaja: TkL Vesa Höttä

Tavaralajimenetelmän metsäkoneen datapohjainen kunnonvalvonta on kehittynyt tasolle, jossa huomattava määrä korkealaatuista tietoa on saatavilla harvesterin puunkäsittelyprosessista ja erityisesti harvesteripäältä, joka on harvesterin tärkein toiminnallinen osa. Menetelmät, joilla olennainen informaatio pyritään löytämään datasta, eivät kuitenkaan ole kehittyneet samalla tavalla. Prosessiteollisuudessa käytettyjä menetelmiä on yritetty soveltaa myös metsäkoneisiin, mutta toistaiseksi menestys on ollut heikkoa. Ongelmana on ollut, että ympäristömuuttujien sekä harvesterin kuljettajan vaikutukset puunkorjuuprosessin suorituskykyyn ja tehokkuuteen ovat erittäin suuria. Lisäksi näiden vaikutusten luotettava mittaaminen ei ole vielä ollut riittävällä tasolla.

Tässä diplomityössä esitellään kolme harvesteripään datapohjaisen kunnonvalvonnan menetelmää, joita ei ennen ole käytetty metsäkoneissa. Menetelmiä on käytetty prosessiteollisuuden puolella aiemmin. Yksi käytetyistä menetelmistä on tiheyspohjainen klusterointimenetelmä ja kaksi muuta ovat todennäköisyyspohjaisia malleja nimeltään Gaussilainen sekamalli ja Bayesilainen verkko. Analyysin lähtökohtana on datan jakautuneisuuden tutkiminen, säännönmukaisuuksien etsiminen havainnoista sekä riippuvuuksien etsiminen havaittujen muuttujien väliltä. Edelleen näiden havaintojen pohjalta prosessin tilat, mukaanlukien vikatilat ja niihin liittyvät muuttujat pyritään tunnistamaan.

Työn teoriaosassa esitellään metsäkoneen toiminnan ja työvaiheiden perusteet, data-pohjaisen kunnonvalvonnan peruskäsitteet sekä datapohjaisen kunnonvalvonnan menetelmiä sekä näihin liittyvät tilastollisten monimuuttujamenetelmien perusteet. Kokeellisessa osassa esiteltyjä menetelmiä sovelletaan dataan ja näiden sopivuutta analysoidaan. Yhteenveto-osioissa esitellään tulokset, jotka sisältävät kvalitatiivisia havaintoja sekä suosituksia koskien malleja ja dataa. Keskeisimpänä tuloksena on, että käytetty data ei ole riittävää tässä työssä käytettyjen kunnonvalvontamenetelmien tarpeisiin. Pääasialliset ongelmakohdat sekä ehdotuksia näiden ongelmien poistamiseksi on esitetty. Näitä tuloksia voidaan käyttää tulevissa tutkimuksissa.

Sivumäärä: 75
Avainsanat: metsäkone, data-analyysi, kunnonvalvonta,
tiheyspohjainen klusterointi, Gaussinen sekamalli, Bayesilainen verkko

Täytetään tiedekunnassa
Hyväksytty: **Kirjasto:**

Acknowledgements

Now, when my studies at TKK are approaching the graduation, it stops me to think about all that has lead to this. The many memorable moments that I have been privleged to spent with my friends and associates during this remarkable time have left a lasting impression on me. Among all that is beyond my ability to describbe, it has changed my way of thinking and way of just having a good time with people. Change - that is what graduation truly is, I hear many people say. After almost seven years of studying in the most rewarding community together with the best imaginable people, I can say that I am ready for this change and prepared to face the new challenges that lie ahead, whatever they are.

I prepared this Master's thesis in the Control Engineering Group of the Department of Automation and Systems Technology at Helsinki University of Technology. I want to thank professor Heikki Koivo for giving me this wonderful opportunity to work in his group and the LATU project. Equally, I want to thank my instructor Lic.Sc. Vesa Hölttä for excellent tutoring support, professor Raimo P. Hämäläinen for supervision of my thesis and M.Sc. Kalevi Tervo for technical advice and numerous valuable ideas. Moreover, I am grateful for having all the technical guidance that I have needed from D.Sc. Arto Peltomaa and M.Sc. Aki Putkonen from John Deere Forestry. Equally, I am grateful for William Martin for helping me with my english writing. Also a very special thanks to the whole Control Engineering Group for all their encouragement and support.

Finally, I want to thank my parents Esa and Maria and my sister Tanja without whom none of this would have been possible. Thank you for being there and supporting me in every imaginable manner. A very especial gratitude I want to express to my dear girlfriend Heli, who has given a special purpose for each day of my life.

In Espoo, on April 11th 2009

Contents

List of Notations

α	Statistical significance level
$\boldsymbol{\mu}_k$	Expected value of the kth cluster
$\boldsymbol{\theta}$	Expected value of the cluster center
$\mathbb{R}^n$	n-dimensional real space
$\mathbb{V}$	Variable space, i.e. , the columns space of $\mathbf{X}$
$\mathbb{X}$	Observation space, i.e. , the row space of $\mathbf{X}$
$\bar{\mathbf{x}}$	Mean vector consisting the average of each column of the data matrix
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
$\boldsymbol{\Sigma}$	Covariance matrix
σ^2	Variance parameter
$\mathbf{A}$	Classification data matrix containing the values of non-metric variables
$\mathbf{a}$	Classification variable
$\mathbf{D}$	Index data matrix containing all the data from the forest harvester
$\mathbf{I}$	Identity matrix
$\mathbf{S}$	Sample covariance matrix
$\mathbf{t}$	Observed time stamps, elements in ascending order
$\mathbf{U}$	Orthogonal matrix of eigen vectors as its columns
$\mathbf{W}$	Matrix of coefficient parameters
$\mathbf{X}$	Data matrix containing the values of metric variables
$\mathbf{x}$	Vector of random variables
$\mathbf{X}_c$	Centered data matrix
$\mathbf{z}_k$	Independent latent variables of the kth cluster
$\mathbf{x}_{\cdot j}$, $[\mathbf{X}]_{\cdot j}$	jth column vector of the data matrix
$\mathbf{x}_{i\cdot}$, $[\mathbf{X}]_i$, $\mathbf{x}_i$	ith row vector of the data matrix
ε	The hypersphere radius parameter of the DBSCAN algorithm
$\lvert\cdot\rvert$	Determinant of a square matrix
$A = \{A_j\}$	Set of classification variables related to the observations
C_k	A cluster with index k
$D = \{D_j\}$	Set of (random) variables containing the sets X, A and t
E	Set of edges of a graph
$F(p, n)$	F-distribution with p and n degrees of freedom

G	Directed acyclic graph
K	Number of clusters/components in a cluster/mixture model
n	Number of observations
$N(\mu, \mathbf{\Sigma})$	Normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$
$N_\varepsilon(\mathbf{x}_i)$	The ε-neigbourhood, i.e. , the set of points within distance ε from $\mathbf{x}_i$
N_{min}	The minimum number of points parameter of the DBSCAN algorithm
$p(\cdot)$	Probability density function
t	Timestamp variable
$T^2(p, n)$	Hotelling's T^2 distribution with p and n degrees of freedom
V	Set of vertices of a graph
$W(\mathbf{\Sigma}, n)$	Wishart distribution with covariance $\mathbf{\Sigma}$ and n degrees of freedom
$X = \{X_j\}$	Set of index variables related to the observations
$\mathrm{tr}(\cdot)$	Trace of a square matrix

List of Abbreviations

c.d.f.	Cumulative Distribution Function
i.i.d.	Independent and Identically Distributed
p.d.f.	Probability Density Function
p.s.d.	Positive-Semidefinite
BNET	The Bayes Net toolbox for MATLAB
CM	Condition Monitoring
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EM	Expectation Maximization
FDI	Fault Detection and Isolation (also Fault Detection and Identification)
GMM	Gaussian Mixture Model
MATLAB	Matrix Laboratory (Mathworks, Inc.)
OpenBUGS	A Bayesian inference software
	(BUGS = Bayesian inference Using Gibbs Sampling)
PCA	Principal Component Analysis

CHAPTER 1

Introduction

A major part of today's forest harvesting is perfomed with harvesters involving a substantial amount of automation and information processing. Such harvesters can process an entire tree within a minute and achieve high standards in the efficiency and productivity of the work. The main parts of the harvesting process, however are still controlled by a human operator but the computer aided subsystems provide a huge amount of guidance for the operator. Because of the amount and complexity of the mechanical and electrical components and subsystems, the harvester is quite vulnerable to faults caused by mechanical deterioration and wear. Moreover, the high level of dependence on human actions in the interaction between the harvester and the operator makes the overall process very sensitive to small variations in the practices executed by the operator. These faults and malpractices affect greatly the quality of the work and, therefore, their prevention can improve the cost-efficiency of the forest harvesting process.

Due to development in measurement electronics and data handling electronics in the last decade, it has become possible to implement high-end data processing systems into forest harvesters (see Hölttä (2004) and Hölttä et al. (2005) for details). These systems enable thousands of measurements in a resolution of just milliseconds to be recorded from harvester movements and the different phases of harvesting process. Evidently, collecting data with such accuracy produces a huge amount of data for analysis. The amount and quality of collected data is comparable to the data that are collected from

heavily monitored industrial processes. So, apparently the data analysis and diagnostics methods that have a strong basis in industrial use, e.g. , see Harris et al. (1999) and Chiang et al. (2001), could be applied also in forest harvesting.

Data collected from forest harvesters are already used for online and offline decision making. Online decision making is mostly based on the current data or on a very short history of the data and the relevant decision maker is the harvester operator. Correspondingly offline decision making is based on a longer history of data and is performed by data analysts or specialists. So far, decision making has been manual and mostly based on descriptive statistics. Some fault detection and condition monitoring methods have already been suggested in (Repo et al., 2006) and (Repo, 2008), which apply the fault detection techniques to data with artificial fault cases. Practically, this thesis is building upon the results of these publications.

The aim of this thesis is to examine data analysis methods and statistical methods that could be helpful in interpreting the variations and unexpected changes in data collected from forest harvesters. This thesis is written from the data-analyst point of view and thus all background and supplemental information as well as technical issues related to forest harvesters and forest harvesting in general are explained only briefly to provide the reader with the most necessary information about the source of the measurement data. The focus is especially on the examination of the data driven condition monitoring and fault detection methods that support the decision making of the forest harvester data analyst. Particularly, three different condition monitoring methods are introduced and their applicability to the data is examined. These methods are *density based clustering analysis*, *mixture models* and *Bayesian networks*. The motivation for using each of these models is first of all that they have been previously used very widely in monitoring the condition of industrial processes, e.g. , see Lee et al. (2004), Choi et al. (2005), Detroja et al. (2006) and Lerner et al. (2000). There are, however, a vast number of additional methods used in monitoring industrial processes also. So, the reason for choosing these methods in particular is that one of the initial assumptions of the forest harvester experts is that the harvester operation is divided into states, called operations points, where the

harvester operation differ significantly from the operation of the other states. For this reason, the chosen methods are considered to be useful in finding these assumed states of the harvester operation and to discover the factors that mostly vary between these states. The examined methods are expected to be able to distinguish the operation that can be explained by a known variation in environmental condition from the operation which cannot be recognized as a consequence of any known change in the surrounding conditions and is, therefore, interpreted as a fault operation.

This thesis consists of three parts: a theoretical part, an experimental part and the final conclusions and discussion. The theoretical part starts with an introduction to the data-driven fault detection methods in Chapter 2, the essential background information and basic terminology of forest harveter operation is given in Chapter 3 and an introduction of the measurement system and description of the index data are given in Chapter 4. The final sections of the theoretical part consider general multivariate statistical methods on the index data in Chapter 5 and give an introduction of the three used condition monitoring methods in Chapter 6. The experimental part is in Chapter 7 which contains an examination of the available data and applies the introduced models to the data. Finally, a summary of the results together with the final conclusions and proposals for areas of further studies are given in Chapter 8.

CHAPTER 2

Condition Monitoring

Increasing requirements in productivity, efficiency and environmental sustainability have pushed systems engineers to develop reliable methods to monitor and control industrial processes as well as the processes of forest harvesting, agriculture, mining etc. Faults and malfunctions in highly automatized and complex systems can be hard to detect and remove. Multiple condition monitoring (CM) methods have been developed and applied for processes in the process and manufacturing industries but their applicability to forest harvesting processes have not been attempted to the same extent. This chapter introduces the main concepts and approaches of process condition monitoring and fault detection.

2.1 Fault Detection and Identification

The aim of process condition monitoring is to detect any kind of undesired disturbances or faults in process variables and properties and finally to remove them. Chiang et al. (2001) have represented process monitoring as a loop with four procedures: *fault detection, fault identification, fault diagnosis* and *process recovery* (the used terminology is adopted from (Raich and Çinar, 1996)). A block diagram of the process control loop is shown in Figure 2.1. Fault detection determines whether the abnormal behavior in the process is a fault or not. Fault identification identifies the process variables that differ most relevantly when the fault occured. Thus, it determines the subsystem or part of

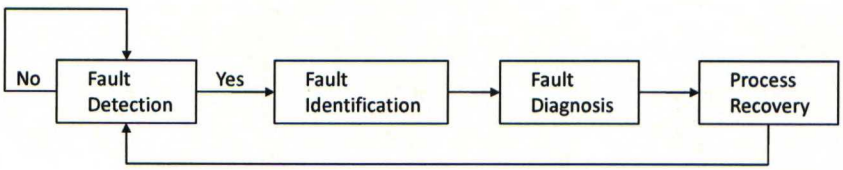


Figure 2.1: A schema of the process monitoring loop presented in Chiang et al. (2001).

the process where the fault occured as precisely as possible. Fault diagnosis determines the reasons why the fault occured and futhermore the type, amount and time of the fault. Finally, process recovery is the action taken to remove the fault and return the system back to normal operation. In this thesis, the main attention is on the first two procedures, although the reasons for faults are discussed qualitatively.

From the condition monitor point of view it is reasonable to define two states of the examined system: the *in-control status* is a normal state of opertation where all system variations are either predictable or within permitted limits, the *out-of-control status* is an abnormal state of the system where some variable, or several variables, have unpermitted values due to faults in the system.

2.2 Process Monitoring Approaches

Process monitoring mostly utilizes methods based on statistical theory, pattern recognition methods, information theory and the theory of dynamic systems. These methods are used to extract information from the process states, structure and behavior. In developed process monitoring schemes, these methods are combined to obtain more thorough methods, for example, different subsystems can be modeled using different methods. Process monitoring methods are usually classified into three approches, namely *data-driven*, *analytical* and *knowledge-based* (Chiang et al., 2001). A data-driven approach utilizes directly the process data available, assuming only a little information about the dynamical structure or functionality of the process. In statistical terms, the data-driven approach concentrates on finding the probability distribution of the data.

This is mainly obtained by statistical methods and pattern recogniton methods. The results obtained by the data-driven methods can be only as good as the quality of the data used. An analytical approach is more extensively based on the process model. The process in- or out-of-control states are determined by comparing the measured data to the reference data given by the model. Thus, the model already defines the basic structure of the probabilty distribution and the statistical methods can be used to analyze the residuals - the difference between the modeled and measured data. Unfortunately, the model construction can become very difficult in the case of complex systems, which reduces the usability of analytical methods. Knowledge-based methods mainly rely on the qualitative information and the process specific information that is provided by the system experts. The process modeling is based on human-oriented rules and logic. However, information provided by experts can be very difficult to model mathematically or by any change deterministically. A very widely used knowledge-based method is *fuzzy-logic*, which is based on the theory of fuzzy sets. Fuzzy-logic is a very useful method for modeling imprecise, undeterministic or even subjective information of the system experts.

The scope of this thesis is mainly on the data-driven approach utilizing also methods that are principally knowledge-based. An analytical approach is beyond the scope of this thesis mainly because there exists no adequate model of the harvesting process or the harvester head procedures.

2.2.1 Data-driven Approach

The core idea of data-driven process monitoring is to characterize the variations in the process data and to interpret the reasons for variations. Principally, there are two types of variations in the process that can be seen in the process data, namely *common cause* and *special cause* (Ogunnaike and Ray, 1994). Common cause variations are entirely randomly caused and often better known as random noise, whereas special cause variations are all the variations that are not explained by random variations. Common

cause variations cannot be predicted because of their randomness, only confidence intervals for common cause variations can be determined. Special cause variations are usually connected to the process states and, therefore, they can be predicted as the process state change. Fault detection with data-driven methods is closely related to the recognition of these two types of variation in the process data. Distinguishing a normal variation from an abnormal variation is practically the same as extracting the two above-mentioned types of variation from the data. These methods rely on the assumption that the characteristics of the variations remain relatively static in a system operating in an in-control status. On the contrary, a fault in the system causes abnormal variation and covariation in the data and, thereby, can be detected by comparing it to the corresponding parameters of the in-control status.

Further, the data-driven approach can be categorized into two approaches: *supervised learning* and *unsupervised learning* (Chiang et al., 2001). In the case of the fault diagnostics, supervised learning means that the fault states of the system are known and described, i.e. each observation from a process in fault state is labeled. Such data is also said to be *complete data.* The model is then fitted to the complete data such that some error criterion is minimized. In the unsupervised learning scheme the presence of the failures are not known and the fault states must be explored from the *incomplete data* using the methods described later. In this thesis the unsupervised learning scheme is mainly used because only incomplete data is available, i.e. there are practically no observations of the assumed states of the system.

2.2.2 Knowledge-based Approach

In the most simple case, knowledge-based fault detection methods are based on qualitative knowledge of the target process. Generally, knowledge is determined as information that is partially uncertain or subjective because it is dependent on the experience and understanding of the observer. Traditional knowledge-based fault detection systems have been implemented as a causal tree model connecting the possible fault to the ob-

served system variables. Examples of such systems presented in Chiang et al. (2001) are *signed directed graphs* and *symptom tree models.* A more sophisticated class of knowledge based methods are *expert systems*, which are more close to human problem solving. Usually an expert system consists of a rule base and an inference engine. The idea of expert systems is to encode human knowledge into a form of logical rules that relate the observed symptoms to the process faults. Expert systems often include other techniques such as *fuzzy logic*, *neural networks* and *pattern recognition* (Chiang et al., 2001).

In this thesis the knowledge-based approach is used together with probabilistic networks represented more precisely in Chapter 6. In particular, this means that the fault-symptom network is constructed by using the same methods as in traditional knowledge-based fault detection but the inference is implemented by modeling the joint probability distribution of the variables involved.

CHAPTER 3

Forest Harvester and Operating Conditions

To familiarize the reader with the origins and the background of the objectives of this thesis, the basic architecture of a forest harvester is explained. To understand the variations and behavior that might occur in the data collected from forest harvesters it is essential to understand the origins of the data and the factors that might be influential. This chapter gives an introduction to the functionality and operation of a forest harvester as well as a description of the main phases of harvesting and mechanical tree processing. In addition, the influences resulting from changes in the working environment as well as the experience and skills of the driver are equally important to keep in mind when considering the total forest harvesting process.

3.1 Architecture and Operation

A forest harvester is constructed on a mobile platform that enables it to move and operate in a fairly difficult terrain. The main architectural parts of a harvester are *harvester head, crane, cabin and controls* and *engine and power transmission.* A schematical picture of a forest harvester is shown in Figure 3.1. In addition to the parts in Figure 3.1, there are measurement and control electronics located at different points on the

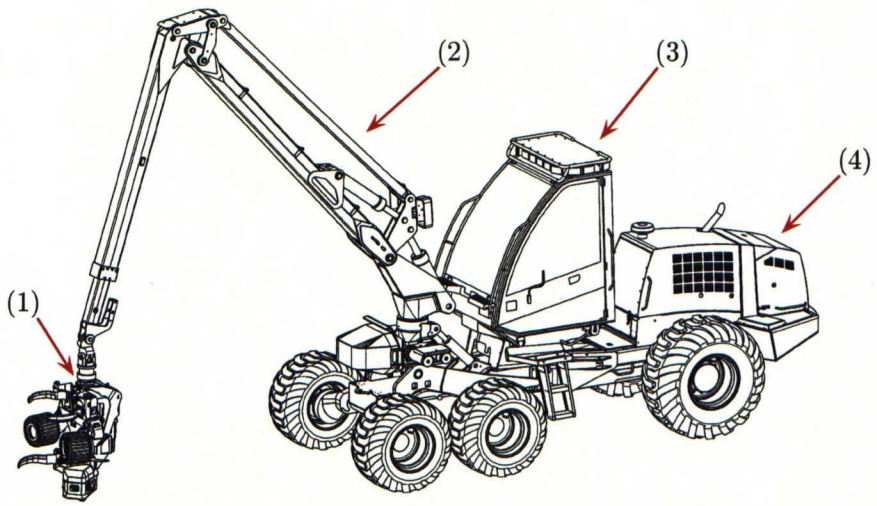


Figure 3.1: Main parts of a forest harvester: (1) harvester head, (2) crane, (3) cabin and controls, (4) engine and power transmission. Adapted from (Timberjack 1).

harvester, that principally enable the automatical operation procedures in the harvesting process. In particular, this means the operation procedures that the machine can complete autonomously without the direct intervention of the operator, which makes the measurements more independent of the operator. They also increase the safety in the harvesting operation.

The operation of a harvester consists of four main phases (or procedures) namely: (i) felling a tree, (ii) delimbing the tree, (iii) cutting the tree into logs and (iv) proceeding to the next tree. The most important part involved in the mechanical handling of the wood and that which especially affects the efficiency and quality of the harvesting process is the harvester head. The main parts of a harvester head are the *delimbing knives*, *feed rollers*, *chain saw* and the *measurement sensors*. A schematical figure of a harvester head is shown in Figure 3.2.

Three of the above-mentioned main operation phases take place particularly in the harvester head. These phases are explained more precisely in the following. (i) In the felling phase harvester operator grabs a tree with the harvester head and cuts it with the chain saw. The tree is then felled in the desired direction by using the crane. The

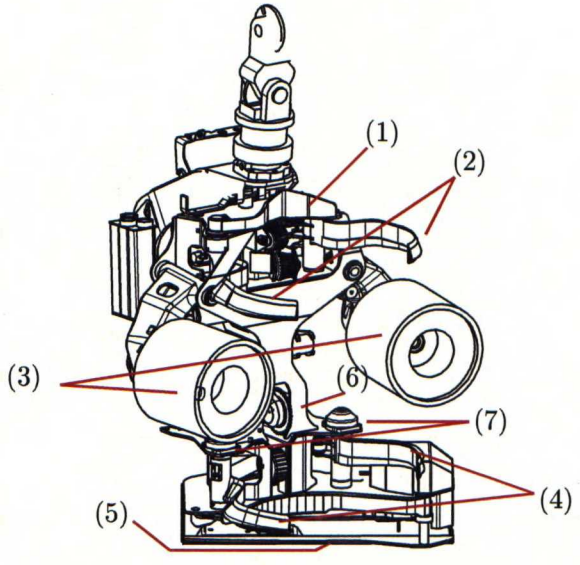


Figure 3.2: Main parts of a harvester head: (1) delimbing knife (fixed) (2) upper delimbing knives (3) feeding rollers (4) lower delimbing knives (5) chain saw (6) length measurement wheel (7) diameter sensors. Adapted from Timberjack 2; Hölttä (2004).

harvester head is placed as close to the root of the tree as possible to gain maximum profit from the tree and for leaving as low stumps in the forest as possible. (ii)-(iii) The delimbing and cutting phases take place simultaneously. Immediately after the felling phase a subprocedure named *feeding* starts to feed the stem toward a suitable position for cutting. While the stem proceeds between the feeding rollers, the branches are removed with the delimbing knives. When the stem has reached the cutting position successfully, the chain saw cuts a log which drops out of the harvester head. Feeding starts again and the operations in phases (ii) and (iii) are repeated until the whole stem is processed. After the last properly sized log is cut, the top of the tree is discarded.

In phase (iv), the harvester operator chooses the next tree, proceeds to it and starts over from phase (i). The harvester head is not used in phase (iv). So, it is evident that the fourth phase has no direct influences on the operation that take place in the harvester head. However, it can affect indirectly, because for example, the position and the orientation of the harvester might influence the possibilities of the operator to process the next tree without obstruction.

Modern forest harvesters are equipped with several sensors that measure the most essential physical quantities related to the harvester operation. Sensors are placed in all parts of the harvester and they make measurements during the different phases of the operation. Some examples of the sensors are shown in Figure 3.2. All sensors are not necessarily visible but integrated into the harvester control electronics. Such measurements are mostly related to the time consumed by a particular operation phase or between subsequent operation phases or otherwise the number of repetitions required to complete an operation phase. More details about the measurement data will be given in Chapter 4.

3.2 Operating Environment

There are numerous influential factors in the operating environment of a forest harvester especially when compared to strictly isolated industrial processes where the surrounding conditions can be monitored very easily and even controlled quite arbitrarily. Different types of forest, trees and terrain or variations in climate and temperature are only a few of the influential factors that are not monitored at all so far. And not to mention, controlling them externally is totally beyond reach. Moreover, the effect of the above mentioned factors can be mixed quite complexly such that the influences would be quite hard to predict even though some of the factors could be measured. For instance, rainy or humid weather can make the harvested trees more slippery which affects the feeding in the harvester head. These weather conditions make the terrain softer and more difficult to proceed. The time of the year and even the time of the day can be influential affecting both the mechanical performance of the harvester and the physical properties of the working environment.

Such complex operating conditions give rise to very noisy sensor measurements in forest harvesters. Measurements taken under different conditions are not necessarily comparable to each other if there is no information about the variations of the environmental factors. This is important to keep in mind when performing data analysis on measure-

ment data. It is not always possible to extract the influences of all environmental factors from the measurement data. Especially this brings difficulties to fault detection and identification, because it is difficult to distinguish faults rising from changes in operating conditions from faults rising from machine failure.

3.3 Contribution of the Operator

The skill and experience of the harvester operator naturally has a huge influence on the efficiency and productivity of the work. An experienced operator can be 40% more productive than an inexperienced operator (Väätäinen et al., 2005). The operator can affect different work phases by using different actions and methods that are characteristic to virtually every operator. Generally, however, harvester operators follow predetermined instructions in their work that guarantee a satisfactyory level of performance. These instructions are better known as *work techniques* (Ovaskainen, 2009). They consist of the basic routines and techniques that are required in handling and moving of the tree. The differences between inexperienced and experienced operators can be seen especially in the fine-tuning of these routines and the ability to combine the subsequent phases smoothly. These personal skills are often referred to as *tacit knowledge*, which means all the knowledge that an operator has adopted but is difficult to put into words. These work techniques and this individual knowledge has a great impact on the performance of the harvesting process. Two simple and clarifying examples of this could be, firstly, the starting of the feed while the tree is still falling and, secondly, the opposite crane movements during the feeding. The first example eases the work of the feeding rollers, because when the tree is still in an upright position, the gravitational force will help it to move between the rollers, speeding up the initial acceleration. The second example, the opposite crane movement during the feeding, has a very similar effect, meaning that the operator moves the crane to the opposite direction of the feed when the feed phase begins. Thus, the initial acceleration and, furthermore, the average feeding speed of the tree will both be improved.

Another factor affecting the productivity of the work is the machine *parameter settings.* There are several parameters that especially influence the operation in the harvester head (Timberjack 1). These are, for example, *feed currents*, notably the *maximum feed current* and *saw pressures.* The feed currents determine the maximum acceleration and speed of the feeding. Thus, a greater current gives a greater maximum feeding speed and acceleration. If the operator is not skilled, however, the greater feeding speed often decrease the quality and efficiency of the work. The saw pressures determine, for instance, the maximum pressure by which the saw flange is pressed againjst the tree which has a huge influence on the performance of the saw. Typically, the parameter settings mentioned here are personal preferences for each individual operator.

CHAPTER 4

Measurement Data and Indices

The harvester head is perhaps the most essential part in the harvesting work, with regard to the quality and efficiency of the work. It is, therefore, reasonable to take the harvester head measurements under special consideration. To achieve reliable overall analysis based on the harvester head measurements, it is essential that the measurements from different machines and operational conditions are comparable with each other. This chapter describes the operations that take place when the raw measurements from the harvester head are converted into a more sensible form using proper classifications and transformations.

4.1 Raw Measurements

The forest harvester data processing system stores the measurement data as records that contain the processing data of a single stem. These measurements are mostly physical quantities, processing times or other logical information of the machine operation state or the processed stem. The values of the measurements are either integer or decimal numbers that indicate the quality or quantity related to a single work phase. In general, the measurement data can be classified into four *measurement scales* originally suggested by (Stevens, 1946). These measurement scales are *nominal*, *ordinal*, *interval* or *ratio* scale. Some examples of the measurement quantities are: processing time used to complete a work phase, dimensions of the processed stem, success of a procedure and

number of repetitions required to complete a procedure. These raw measurements require quite an amount of manipulation to be understood by the harvester operator or a data analyst.

4.2 Performance Indices

Measurements do not merely describe the performance of the harvester very well. This is because the desired value of the measured quantity might be uncertain or depend on the values of other measurements or on the operating conditions. Thus, it is desirable that the determination of the measures of performance take into account the operating conditions. This is obtained by calculating the summarized, averaged, or by other means, transformed values from the measurement data. The measurements of a single stem are transformed into *indices* that indicate the performance or success of the stem processing, making them more understandable to a data analyst. It is convenient that the indices are connected to the true performance of the harvester and, moreover, that the indices would be easily comparable with each other. Therefore, determination of an index combines more than one measurement from the harvester head. Determination of the indices is explained thoroughly in (Hölttä et al., 2005). The key idea is that the time series of the measurements are split into sequences of 100 stems. For each sequence, the outliers are removed and the measurements are classified into bins that depend on the operating conditions. Therefore, the measurements in each bin should be comparable with each other. Index values are determined for the measurements in each bin separately. This is done by scaling measurements such that the measurements at a predetermined upper bound are assigned to a value of 100 and measurements at a predetermined lower bound are assigned to a value of 0. Finally the index value is calculated as a weighted sum of the index values at each bin. Hence, 100 subsequent measurement values are compressed into a single number, the index of the current variable, that characterizes the desired property.

The specifications for the indices used in this thesis are equivalent to those explained

above with one exception. The index values that are saturated to either 0 or 100 will be replaced by the original values. Initially, all index values were restricted to a 0-100 interval for convenience. However, some of the indices are distributed significantly over the 0-100 interval, resulting in distributions with high densities for values 0 and 100, because all values less than 0 and greater than 100 will be assigned to 0 and 100 respectively. Removing the saturation of indices will produce more descriptive data analysis and more natural statistical models.

4.3 Data Set-up and Notations

The data available in the analysis of this thesis consists of 23 variables and more than 100,000 data points. Because originally each data point has been calculated using 100 subsequent observations from a forest harvester, the data contains measurements from over 10 million trees recorded by more than 150 forest harvesters spreading over a total time period of three years.

In addition to the measurement data obtained from the harvester head and from other parts of the harvester, there is also some qualitative information stored from the harvesting process. These are the *timestamp*, the *harvester ID*, the *harvester model* and the *tree types*. So, each index datapoint is calculated using the qualitative information and the measurement data. The diagram in Figure 4.1 illustrates how the measurements are converted into the indices.

The index data is organized into a partitioned matrix

$$\underset{n\times q}{\mathbf{D}} = \left[\underset{n\times p}{\mathbf{X}} \,\middle|\, \underset{n\times r}{\mathbf{A}} \,\middle|\, \underset{n\times 1}{\mathbf{t}} \right], \tag{4.1}$$

where the columns of $\mathbf{X}$ correspond to the index variables $X = \{X_j\}$, the columns of $\mathbf{A}$ correspond to the classification variables $A = \{A_j\}$ and the column vector $\mathbf{t}$ includes the time stamps of each row. The rows of the matrix $\mathbf{D}$ correspond to n observations.

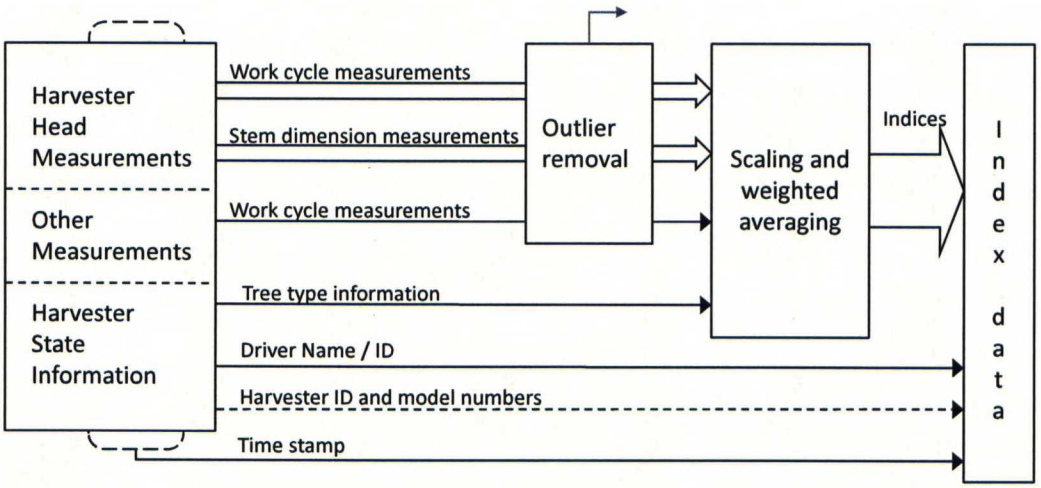


Figure 4.1: Handling of measurements to calculate indices.

The elements of the matrix $\mathbf{X}$ are metric and either interval or ratio scaled variables. Whereas the elements of matrix $\mathbf{A}$ are nonmetric and either nominal or ordinal scaled. The timestamp t_i at the row i tells the time when the row was recorded.

It is useful to determine a few more notations for the matrix $\mathbf{X}$ of the metric variables. Columns $\mathbf{x}_{\cdot j}$ of the $\mathbf{X}$ are vectors of the *variable space* $\mathbb{R}^n$ and the rows $\mathbf{x}_{i\cdot}$ of the $\mathbf{X}$ are vectors of the *observation space* $\mathbb{R}^p$. For convenience, the observation space and the variable space are denoted by $\mathbb{X}$ and $\mathbb{V}$ respectively. The row vectors are also denoted simply by $\mathbf{x}_i$ if there is no possibility of confusion. In addition, some descriptive statistics for the metric data in $\mathbf{X}$ are useful to determine. The mean of the $\mathbf{X}$ is given by

$$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}^T\mathbf{1} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T, \tag{4.2}$$

where $\mathbf{1}$ is $n \times 1$ column vector of ones. The centered data matrix is given by

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^T \tag{4.3}$$

and the unbiased estimate of the covariance matrix is given by

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}_c^T\mathbf{X}_c. \tag{4.4}$$

4.4 Index Data Prehandling

Many statistical models assume the input data to have a certain distribution. Deviations from the assumed distributions can cause problems in the estimation of the model parameters. These deviations are mainly caused by two reasons: *outliers* and *anomalies*. Outliers are observations that are significantly different from other observations in respect to some properties. They can be produced, for instance, by a measurement error or a fault in the system. Usually outliers occur one at a time referring to a totally random measurement error. Anomalies are deviations in the data that are not explained by the model. Occurence of several outliers consequently usually refers to an anomaly which is more probably a fault in the system. An anomaly appears in the distribution as skewness or as multiple modes and usually they directly reflect an insufficiency in the used model.

Generally, any real data contains anomalies and outliers that conflict with the preassumptions of the model. These defects are normally connected either to the variables or to the observations. Variables that are badly distributed (e.g. due to extremely large measurement errors) or are otherwise assumed to have no influence on the issues under examination are removed from the analysis. There are multiple analytical model selection methods such as *adjusted R-squared*, *Bayesian information criterion* or *Akaike information criterion* that can be used to select the number of variables in the model. These methods are usually combined with stepwise methods that select variables by including or excluding variables in a step by step procedure. Similarily numerous techniques for outlier removal are known in the statistical literature. In most cases, outlier

removal techniques are model specific and a few of them are introduced in the next section.

Many parametric statistical methods are not scale invariant. This means that the units in which the data is represented affect the parameter estimates. The input variables with essentially different scalings affect the output variables and the results such that variables with greater values will dominate. This can be seen for example in the principal components analysis method (see Section 5.2) where the total variance is the sum of the variances of the individual variables. One variable having a significantly greater variance will be dominating the total variance which causes the model to give greater weighting for this variable. The incorrect weighting of the variables can be prevented by *scaling* the input data properly. This can be done, for example, by unit variance scaling, which is obtained by dividing each column vector of the centered data matrix $\mathbf{X}_c$ by the its variance. If the information about the variance is not available, the sample variance given by the diagonal elements of $\mathbf{S}$ can be used as well. The index data is already scaled approximately to a 0-100 interval. Thus, it would be reasonable to assume that the indices are already properly scaled. This assumption is initially trusted in the following sections, but if this scaling proves to be insufficient, a proper scaling will be considered. In the rest of the thesis the observations in $\mathbf{X}_c$ will be properly scaled unless mentioned otherwise.

CHAPTER 5

Multivariate Statistics on Index Data

Statistical inference and methods are essential parts of almost any sophisticated data-driven fault detection technique. Univariate *statistical process monitoring* as well as its extension the *multivariate statistical process monitoring* both rely strongly on statistical methods and on decision making based on different statistics obtained from data. Numerous advanced methods, for example, *pattern recognition* and *Bayesian methods* have their foundations in classical statistical methods. This section introduces several statistical techniques that are important in the data-driven methods used in this thesis.

5.1 Outlier Removal with Hotelling's T^2-statistic

Assuming the data to be normally distributed with a mean value $\boldsymbol{\mu}$ and sample covariance matrix $\mathbf{S}$ the outliers can be removed using the following statistic. (Mardia et al., 1979)

Theorem 5.1. (Hotelling's T2-statistic) *Given i.i.d.* $p \times 1$ *data sample* $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

and a sample covariance matrix $\mathbf{S}$ *such that* $(n-1)\mathbf{S} = \mathbf{M} \sim W(\mathbf{\Sigma}, n)$, *then*

$$T^2 = \frac{n}{n-1}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{S}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \sim T^2(p,n), \tag{5.1}$$

where $T^2(p,n)$ *is Hotelling's* T^2 *distribution with* p *and* n *degrees of freedom that is equivalent to* $\frac{np}{n-p+1}F(p, n-p+1)$, *where* F *is the* F*-distribution.*

Proof The proof for theorem 5.1 is given in the Section 3.5. of Mardia et al. (1979).

Given the distribution of the test statistic T^2, a threshold value for outlier removal can be determined. At $100(1-\alpha)\%$ confidence level the threshold value is

$$T^2_\alpha = \frac{np}{n-p+1}F_\alpha(p, n-p+1), \tag{5.2}$$

where F_α is the determined as $\Pr(X < F_\alpha(p, n-p+1))=1-\alpha$ that is the integral of the c.d.f. to the critical upper level. An example of the T^2-statistic in a two dimensional case is shown in Figure 5.1.

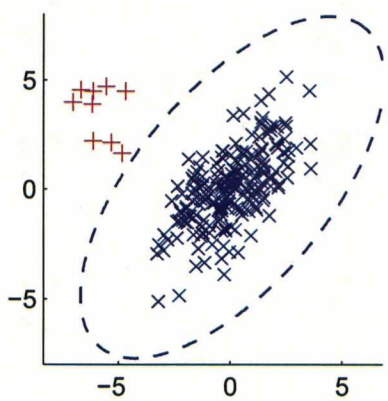


Figure 5.1: Acceptable data points '×' separated from outliers '+' using $\alpha = 0.1$. Points inside the dashed line are within the threshold $T^2_\alpha = 4.66$.

5.2 Dimension Reduction with PCA

Principal Component Analysis (PCA) is a multivariate technique that is used for reducing the number of variables in multivariable data. In PCA the original variables $\mathbf{x}$ are transformed into a set of new uncorrelated variables using linear transformations. The key idea of the principal component analysis is to find such direction $\mathbf{u}_1$ in the observation space that the variance of the data along that direction is maximized. New directions $\mathbf{u}_i$ are determined subsequently such that they are orthogonal to the previous directions but capture the maximum variance in the remaining directions. Altogether, p such directions can be determined. This is also known as the *maximum variation formulaiton* of the PCA (Bishop, 2006) and the relevant optimization program can be formulated as

$$\begin{aligned} \max_{\mathbf{u}_i} \quad & \mathbf{u}_i^T \boldsymbol{\Sigma} \mathbf{u}_i \\ \text{s.t.} \quad & ||\mathbf{u}_i|| = 1 \\ & \mathbf{u}_i^T \mathbf{u}_j = 0 \qquad \forall j < i, \end{aligned} \tag{5.3}$$

where $\boldsymbol{\Sigma}$ is the positive-semidefinite (p.s.d.) covariance matrix of the random vector $\mathbf{x}$. The solution to (5.3) is given by the *eigenvalue decomposition* of $\boldsymbol{\Sigma}$ that is

$$\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T, \tag{5.4}$$

where $\mathbf{U} = [\mathbf{u}_1 \ \dots \ \mathbf{u}_p]$ has the eigenvectors as its columns and for which holds $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$ is a diagonal matrix of the eigenvalues. The diagonal elements of $\boldsymbol{\Lambda}$ are ordered in descending order such that $\lambda_1 > \lambda_2 > \cdots > \lambda_p$ and the eigenvalue λ_i corresponds to the variation in the direction of $\mathbf{u}_i$. The new variables called *principal components* are now obtained by $\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$. It is easy to show that $\text{E}(\mathbf{z}) = \mathbf{0}$ and

the covariance for $\mathbf{z}$ is thereafter given by

$$\begin{aligned}
\mathrm{Cov}(\mathbf{z}) &= \mathrm{E}[(\mathbf{z} - \mathrm{E}[\mathbf{z}])(\mathbf{z} - \mathrm{E}[\mathbf{z}])^T] = \mathrm{E}[\mathbf{z}\mathbf{z}^T] \\
&= \mathrm{E}[(\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}))(\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}))^T] \\
&= \mathbf{U}^T\mathrm{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]\mathbf{U} \\
&= \mathbf{U}^T\boldsymbol{\Sigma}\mathbf{U} = \mathbf{U}^T\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T\mathbf{U} = \boldsymbol{\Lambda},
\end{aligned} \tag{5.5}$$

which shows explicitly that the principal components $\mathbf{z}$ are uncorrelated and their variances are equal to the eigenvalues λ_i. Hence, the sum of the eigenvalues $\mathrm{tr}(\boldsymbol{\Lambda})$ can be used as a measure of the total variation in the data and it is equal to $\mathrm{tr}(\boldsymbol{\Sigma})$. The essential question when applying PCA is that how many principal components describe the data sufficiently. Answer to this question can be found by inspecting the eigenvalues. The number of pricipal components that capture a sufficient amount of variation in the data is determined by summing the eigenvalues up to λ_k until a sufficient proportion of the total variation $\mathrm{tr}(\boldsymbol{\Lambda})$ is covered. There are multiple heuristics for choosing a proper number k of principal components. Few of them are represented by Jackson (2003) and Sharma (1996). The most intuitive is the *SCREE test* in which the eigenvalues are plotted in descending order of magnitude and the k is chosen at an elbow where the degree of decrease changes most. Applying the PCA to real data, the covariance matrix $\boldsymbol{\Sigma}$, the expected value vector $\boldsymbol{\mu}$ and the random variable $\mathbf{x}$ are replaced by the sample covariance matrix $\mathbf{S}$, the sample mean $\bar{\mathbf{x}}$ and the observations, i.e. , the rows of $\mathbf{X}$, respectively.

CHAPTER 6

Data-driven Condition Monitoring Methods and Models

The starting point for the data-driven fault detection was introduced in Chapter 2. The in-control status, the out-of-control status and the connection between the different kinds of variation in data and the system faults were determined as well as the unsupervised learning scheme that will be used in the models of this thesis. Data-driven condition monitoring relies basically on the above-mentioned concepts. The objective of the data-driven fault detection methods used here is to find *latent states* and *latent variables* in the data that express how each part of the system is involved in the fault occured. In this case, the latent states of the system refer to both the in-control and out-of-control states of the system. The applicability of these models is strongly dependend on the distribution of the data, or more particularly, on the methods that model the distributions.

In this thesis three approaches are applied, namely *data clustering model*, *mixture models* and *Bayesian networks*. In the data clustering model, the latent states of the system are separated using the clustering methods introduced in this chapter. The clustering model is improved by statistical methods that describe the distributions of the individual states. This model is better known as a *mixture model*. A Bayesian networks are well developed methods that are mainly used in decision analysis and pattern recogni-

tion. Bayesian network is usually illustrated as a directed graph of causal relationships. Basically, it is a probabilistical model that takes advantage of the Bayesian inference applied on the hierarchical model of the process variables. The Bayesian model constructed here is a combination of the knowledge-based and data-driven approaches. This chapter introduces the theoretical background of the above-mentioned models.

6.1 Data Clustering

The aim of a data clustering method is to group the n data samples $\mathbf{x}_i$ into K homogenous classes, or *clusters*, denoted by C_k, and where K is usually much smaller than n. Each cluster contains observations that are similar to each other in respect to the properties (the variables) of the data. On the other hand, the samples in different clusters should be as dissimilar as possible. The total number of observations in cluster C_k is denoted by n_k and the belonging of an observation $\mathbf{x}_i$ to cluster C_k is denoted by $\mathbf{x}_i \in C_k$. Data clustering is very helpful in data prehandling when there is no information about the anomalies in the data yet but the data points are assumed to be condensed into several locations in the observation space. The different locations can be diagnosed to belong either to the in-control status of the system or to the out-of-control status of the system. Thus, separation of the clusters plays an essential role in determining the fault states in the measurement data. In this section, a clustering method called Density-Based Spatial Clustering of Applications with Noise *DBSCAN* is introduced, which is very useful in clustering the index data. The section also presents the argumetns why the basic clustering methods *hierarchical clustering* and *partitioning clustering* are inadequate in the case of the index data.

6.1.1 Hierarchical and Partitioning Clustering

Hierarchical clustering methods order the data points into a distance tree called a *dendrogram* on the basis of the sample distances. The application of hierarchical clustering

methods requires some measure of distance to determine the proximity matrix of the nearness of each pair of data points. Such measures are, for example, *Euclidean distance* and *Mahalanobis distance.* The dendrogram is created either by subsequently merging the two nearest data points (agglomerative approach) or by dividing the clusters into two subclusters (divisive approach) in each step. The hierarchical methods are named after the used distance determination method, for instance, single-linkage, complete linkage, average linkage etc. The termination of the linkage is determined by the critical distance D_{min} between the clusters so that when the distance between the clusters is more than the D_{min}, the clusters are no more merged. (Sharma, 1996; Mardia et al., 1979)

The partitioning methods are based on optimization of a clustering criterion. The algorithm starts with an initial clustering that is based on the prior information about the clustering, for example, the expected number and locations of the clusters. The observations are then moved between the clusters so that the optimization criterion is minimized. Examples of partitioning methods are the *k-means* and *k-medoid* algorithms. Lee et al. (2004) have applied the k-means method in multivariate process monitoring.

The hierarchical and partitioning clustering methods have some generally known drawbacks that make them inappropriate for use in index data clustering. Firstly, the computational costs increase quite dramatically when the amount of datapoints n increases. For example, determination of the $n \times n$ proximity matrix in the case of hierarchical clustering requires $O(n^2)$ calculations. Secondly, the methods are vulnerable to noise, i.e. , the single points that do not clearly belong to any cluster in the data. (Ester et al., 1996; Mardia et al., 1979)

6.1.2 Density Based Clustering

Density based clustering methods are used especially when there are clusters with varying density and outliers in the data. DBSCAN (Ester et al., 1996) is a very useful algorithm for performing density based clustering. Some variations of the DBSCAN are

known in the literature, for instance, by Ankerst et al. (1999) and Kriegel and Pfeifle (2005), however the original algorithm is sufficient in the scope of this thesis.

Application of the DBSCAN algorithm requires a few definitions. In DBSCAN, two parameters, N_{min} and ε, are determined beforehand. N_{min} is the minimum number of points that has to belong to the neighborhood of a point. The neighborhood of a point is the hypershere of the radius ε around the point. The set of points within the distance ε from the point $\mathbf{x}_i$ is called the *ε-neighborhood* and is denoted by $N_\varepsilon(\mathbf{x}_i)$. A point $\mathbf{x}_1$ is *directly density-reachable* from point $\mathbf{x}_2$ if $\mathbf{x}_1 \in N_\varepsilon(\mathbf{x}_2)$ and there are at least N_{min} points in the $N_\varepsilon(\mathbf{x}_2)$. A point $\mathbf{x}_1$ is *density-reachable* from point $\mathbf{x}_m$ if there is a chain of points $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots \mathbf{x}_m$ such that $\mathbf{x}_i$ is directly density-reachable from $\mathbf{x}_{i+1}$.

Using the above definitions the DBSCAN algorithm assigns each point $\mathbf{x}_i$ to a cluster C_k that contains the points that are density-reachable from $\mathbf{x}_i$ using the given parameters N_{min} and ε. The remaining unclassified points are the noise observations that do not belong to any cluster. Principally this is achieved by going through all the points and assigning them to a new cluster or to an existing cluster depending on the above conditions. Details of the algorithm are presented in Ester et al. (1996).

The applicability of the DBSCAN algorithm is demonstrated with simulated data. The simulation data contains five components in two-dimensional observation space. Two of these components simulate the in-control states and three of them simulate the out-of-control states. The indices are named as Index 1 and Index 2 and they describe arbitrary performance indices of the forest harvester index data. The data originates from Gaussian distributions. The DBSCAN algorithm was run with parameters $N_{min} = 5$ and $\varepsilon = 4$ and the results are shown in Figure 6.1. The two groups outmost on the right-hand side represent the in-control states and the remaining three groups on the left-hand side represent the out-of-control states. The points marked with a black cross are classified as noise. In this case the clustering succeeds as it was expected to.

DBSCAN clustering is a very powerful method for removing outliers or clusters that do not contain enough measured data samples. In particular, the out-of-control states of

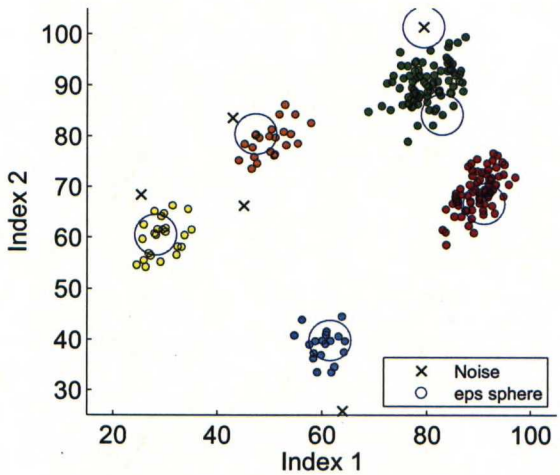


Figure 6.1: Results of the DBSCAN algorithm with simulated data. Used parameter values $N_{min} = 5$ and $\varepsilon = 4$, ε-spheres plotted around a few observations.

the system can produce clusters with only a few points usually because the operation of the harvester is not continued for very long if a fault occurs. However, the in-control states usually contain many data points with high densities and it is, therefore, important to be able to separate these states.

6.1.3 Goodness Criterion for Clustering

The results of the DBSCAN depend on the two adjustable parameters ε and N_{min}. Thus, it is of particular interest to determine the optimal parameter values that produce an optimal cluster configuration with the desired properties. For purposes of fault detection and detecting the in-control and out-of-control states, a suitable optimization criterion would be to form clusters with maximum concentration but in such a manner that the total number of clusters is as small as possible. This problem is a multi-criteria optimization problem. A suitable criterion for maximum concentration of clusters is Mardia et al. (1979)

$$\prod_{k=1}^{K} |\mathbf{S}_k|^{n_k}, \tag{6.1}$$

where $\mathbf{S}_k$ is the sample covariance matrix of the observations in cluster C_k and $|\cdot|$ stands for the matrix determinant. The minimum of (6.1) is determined heuristically in this study. This is because the DBSCAN clustering is not a continuous mapping, which makes the finding of the global optimum very difficult. Using the determinant of the covariance matrix as a measure of the goodness of the clustering is justifiable because the determinant is proportional to the total variation of the data in the specific cluster. Other criteria for clustering are represented in Duda et al. (2000).

6.2 Mixture Model

A general mixture model consists of a discrete probability distribution $p(\mathbf{z})$ and the continuous conditional probability distribution $p(\mathbf{x}|\mathbf{z})$, where $\mathbf{z}$ is a discrete random variable with one component equal to 1 and the others are zeros. The discrete distribution gives a constant probability for each component of the mixture distribution as follows

$$p(z_k = 1) = \pi_k \qquad \forall\, k = 1 \ldots K, \tag{6.2}$$

where $0 \leq \pi_k \leq 1$ and $\sum_{k=i}^{K} \pi_k = 1$. The conditional continuous distributions are the distributions for each component k. The joint distribution of $\mathbf{x}$ and $\mathbf{z}$ is obtained using the product rule of probability as

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}). \tag{6.3}$$

Now applying the Bayes rule for (6.3) gives the conditional distribution of $\mathbf{z}$ given $\mathbf{x}$

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}, \tag{6.4}$$

where the denominator is obtained by

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\mathbf{z}). \tag{6.5}$$

The probability distribution (6.4) is of special interest, because it enables evaluating the likelihood of a new observation $\mathbf{x}_{new}$ belonging to the component k. It is then justifiable to assign the new observation to the component that gives the highest likelihood. (Bishop, 2006)

6.2.1 Mixture Model in Fault Detection

From the condition monitoring point of view, the mixture model seems to be reasonable for modeling a process with discrete state changes. The discrete variables of the mixture model can be interpreted as the operating points of the system. Therefore, mixture models provide a natural probabilistic extension to the model of data clustering.

Applying the mixture model to incomplete data, such as the index data, the distribution of the hidden states, i.e. , the marginal distribution of the discrete variables, has to be learned from the data. Particularly this means estimating the parameters of the probabilities π_i. Estimation of the mixture model is greatly dependent on the specifications of the component distributions. Principally, any multidimensional probability density function can be used as the component distribution if it coincides with the distribution expressed by the data. Although, in the case of only a few distributions, the estimation of the parameters can be carried out easily.

6.2.2 Gaussian Mixture Model

Perhaps the most simple and commonly used continuous component distribution in the mixture models is the Gaussian probability distribution. It is a widely used model for distribution of measurements because it is easy to handle mathematically and it can be

argued by the *central limit theorem*. Another reason for its popularity is the property that many other unimodal distributed data can be converted into a Gaussian distributed variable quite easily. The mixture model with Gaussian component distributions is also known as the *Gaussian Mixture Model* (GMM). Choi et al. (2005) have applied the GMM approach to process condition monitoring. The conditional probability density function $p(\mathbf{x}|\mathbf{z})$ for a GMM can be written as

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(\sqrt{2\pi})^n |\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\Big), \tag{6.6}$$

where the mean and covariance parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ correspond to the k^{th} component of the GMM. Now using (6.5) the conditional distribution (6.4) can be written as

$$p(z_k = 1|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\pi_k}{\sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}. \tag{6.7}$$

Assuming that the observations $\mathbf{x}_i$ are i.i.d. the likelihood function of the parameters given the observed data is the product of the likelihood functions $p(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$. The logarithmic likelihood function can thus be written as

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \Big(\pi_k p(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big), \tag{6.8}$$

where $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ contains the K parameters. The maximum likelihood can be determined by setting the derivative of the (6.8) to zero with respect to the parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. This gives

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{n} \gamma(z_{ik})\mathbf{x}_i, \tag{6.9}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{n} \gamma(z_{ik})(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \tag{6.10}$$

and

$$\pi_k = \frac{N_k}{N}, \tag{6.11}$$

where $N_k = \sum_{i=1}^{n} \gamma(z_{ik})$ and $\gamma(z_{ik}) = p(z_k = 1|\mathbf{x}_i)$ as in (6.7). The derivation of the (6.9) - (6.11) are shown in Bishop (2006). The above equations cannot be solved in closed form, because $\gamma(z_{ik})$ is complexly dependent on the parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. Instead, the representation of the parameters in (6.9) - (6.11) suggests the use of an iterative method for solving the maximum likelihood solution.

6.2.3 Expectation Maximization Algorithm

The parameters of a mixture model can be determined by using the *expectation - maximization* (EM) algorithm (Dempster et al., 1977). The EM algorithm consists of two steps namely the E step and the M step, which are described in more detail for the case of a GMM. The algorithm is represented in the following using (6.9) - (6.11) (Bishop, 2006).

Algorithm 6.1. (EM algorithm for GMM)

1. ***Initialization:*** *Initialize the means $\boldsymbol{\mu}_k$, the covariances $\boldsymbol{\Sigma}_k$ and the discrete probabilities π_k.*

2. ***E step:*** *Evaluate the responsibilities $\gamma(z_{ik}) = p(z_k = 1|\mathbf{x}_i)$ as in (6.7) using the current parameter values π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$.*

3. ***M step:*** *Evaluate new estimates π_k^{new}, $\boldsymbol{\mu}_k^{new}$ and $\boldsymbol{\Sigma}_k^{new}$ for the parameters using equations (6.9) - (6.11) and the current responsibilities. Note: $\boldsymbol{\mu}_k^{new}$ have to be evaluated first to obtain $\boldsymbol{\Sigma}_k^{new}$.*

4. ***Termination:*** *Evaluate the log likelihood using (6.8) and check for convergence. If the convergence criterion is not satisfied return to step 2.*

In practice, the convergence criterion is determined either as a minimum change in the log likelihood or as a maximum number of iterations. The convergence of the EM

algorithm generally depends greatly on the initialization of the parameters, especially if the data is not multinormally distributed or the components are overlapping. The initialization of the algorithm requires some prior knowledge of the data such as the expected number and approximate locations, i.e. , the means, of the components. The correctness of the initialization parameters decreases the required iteration steps of the algorithm. If there is no prior knowledge of the number of components, the algorithm can be executed with a different number of components and the results can be compared with methods described in Section 6.1.3.

The applicability of the GMM and the EM algorithm were tested with simulated data. The used data was the same as used in Section 6.1.2, where more details about the data are explained. The simulated observations are plotted in Figure 6.2(a) with black circles. The two groups outmost on the right-hand side represent the in-control states and the

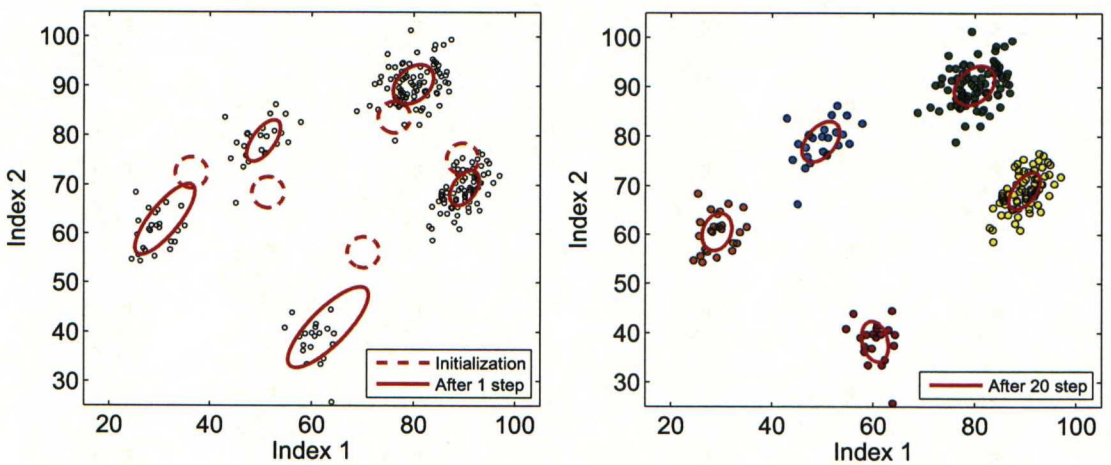


Figure 6.2: Estimation of the GMM with EM algorithm using simulated data: (a) Initialization and first step (b) Convergence after 20 step and final grouping indicated by colors.

remaining three groups on the left-hand side represent the out-of-control states. The progress of the EM algorithm is shown in Figures 6.2 (a) and (b) with red ellipses, that describe the covariance of each component. The dashed ellipses shows the initialization values of the algorithm and the solid ellipses represent the distribution of the components after 1 and 20 iteration steps. The algorithm seem to converge well after 20 iteration

steps. The final belongings of the observations to each component are shown in Figure 6.2(b) with colored circles. This simulation describes the outline of the application of the GMM to the forest harvester condition monitoring. Even though the simulation have been done with simplified data, the main principles of the condition monitoring with real data are very similar.

6.3 Bayesian Network

Bayesian networks, also known as *directed graphical models*, are a part of a probabilistic model family called *probabilistic graphical models*[1]. They incorporate the Bayesian inference and graph theory into a unified formalism that enables building models with high complexity still maintaining the simplicity in description and economical use of parameters. Bayesian networks are used in multiple disciplines of sciences, for example, *decision analysis*, *machine learning* and *pattern recognition*. The applicability of the Bayesian networks is due to their versatility and modularity which enable building complex models by combining simple parts. Bayesian networks cover multiple widely applied statistical models as its special case such as *independent component analysis*, *factor analysis* and different mixture models. An extension of Bayesian networks called a *Dynamic Bayesian network* (DBN) is used to build temporal models that take into account the autocorrelation of the variables. The DBN comprises the probabilistic alternatives for the well-known models of *Kalman filters*, *ARMA models* and *hidden Markov models*. (Friedman et al., 1998)

A formal definition of a Bayesian network requires introducing a few concepts. A *graph* is defined as a pair $G = (V, E)$, where V is a set of vertices (nodes) and E is a set of edges. In G, each vertex is connected to at least one other vertex by an edge. A directed graph is a graph with edges in only one direction and a *directed acyclic graph* (DAG) is a directed graph containing no cycles, that is, one cannot return to a node

[1]Other major class of graphical probabilistic models are the *Markov random fields*, also known as *undirected graphical models*.

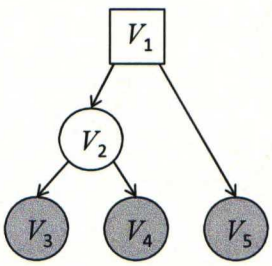


Figure 6.3: A simple Bayesian network of five random variables. Spherical and rectangular symbols refer to continuous and discrete random variables respectively and the shaded variable is an observed variable.

by any path available after visiting another node. In the internal hierarchy of a DAG, it is reasonable to define the parents, descendants and nondescendants of a vertex. For vertices V_j, $V_i \in V$ it holds that

(i) vertex V_j is called a *parent* of the vertex V_i if there is an edge from V_j to V_i, the set of parents of node V_i is denoted by pa_i

(ii) vertex V_i is called a *descendant* of V_j if there is a path from V_j to V_i

(iii) vertex V_i is called a *nondescendant* of V_j if V_i is not a descendant of V_j

A Bayesian network is represented as a DAG, where the vertices correspond to variables and the edges of the graph correspond to relationships or dependencies between the variables. A simple example of a Bayesian network is shown in Figure 6.3. One more definition is needed for the definition of a Bayesian network, namely the *Markov condition.*

Definition 6.1. *(The Markov condition) Suppose a joint probability distribution p of the random variables in set V and a DAG G = (V, E). Together G and p satisfy the Markov condition if for each $V_i \in V$, V_i is conditionally independent of the set of all its nondescendants given the set of its parents. (Neapolitan, 2004)*

Now, a formal definition for a Bayesian network can be given as follows

Definition 6.2. *(Bayesian network) Let p be a joint probability distribution of the variables in some set V, and G = (V, E) be a DAG. Then (G,p) is a Bayesian Network if it satisfies the Markov condition. (Neapolitan, 2004)*

Thus, the joint probability distribution $p(V)$ of the variables in DAG G together with the Markov condition uniquely determine a Bayesian network. Definition 6.2 implies that any conditional and marginal distribution of the variables in V can be represented in terms of the conditional probability densities $p(V_i|pa_i)$ as follows

$$p(V_1, V_2, \ldots, V_n) = p(V_n|pa_n) \ldots p(V_2|pa_2)p(V_1|pa_1). \tag{6.12}$$

Utilizing the joint probability density function p a statistical inference with the Bayesian network can be performed and any Bayesian statistic concerning the variables in the network can be obtained.

To use the Bayesian network for statistical inference with practical problems requires the construction of the DAG G and estimation of the parameters of G. Construction of the model requires some process specific information about the causal relations between the process variables. If there is imperfect information about the relationships of the variables, the model structure can be determined by learning it from the data. This procedure is called the *structure learning* of the Bayesian network which, in principal, is finding the significant relationships between the process variables with respect to proper scoring measures and model diagnostics. The parameter estimation in the context of Bayesian methods is determining the distributions of the parameters, i.e. , the posterior distributions. The parameter distributions are learned from the data by using the *Bayes rule* and the sum and product rules of probability (Bishop, 2006; Gelman et al., 2003). It is often the case that all variables in the network are not observable or are not recorded for one reason or another. The distributions of these so called hidden variables can be estimated also by using the EM algorithm represented in Section 6.2.3.

6.3.1 Fault Detection with Bayesian Networks

Construction of a Bayesian network that determines the causal relationships between the variables makes it appealing to consider whether the different operation points of the system could be added to the model as new variables in the same sense as they were used in the clustering and mixture models. On the basis of the discusison in Chapter 2 about the different approaches of process condition monitoring, it is natural to consider the Bayesian networks as a combination of the data-driven condition monitoring and the knowledge-based condition monitoring. Particularly, the construction of the DAG of a Bayesian network can be seen as the knowledge-based part of the determination of the fault detection model. Similarily, the structure learning of the Bayesian network corresponds to the data-driven part of the construction of a fault detection and diagnosis model.

A few successful examples of applying a Bayesian network for fault detection are represented in Matsuura and Yoneyama (2004) and Lerner et al. (2000). In these studies the structure of the network have been determined from the physical model of the process, which cannot be done in this study, because of the complexity of the process. However, they show that the Bayesian networks are principally very suitable for the fault detection.

In mixture models, the unobserved variables, that were related to the operation points of the system, were modeled as discrete. In the case of complex systems, this might be inadequate and some of the variables should be modeled as continuous, for example, environmental effects vary most probably continuously as well as the parameters of the harvester that can be set to continuous values at a certain interval. A model with both discrete and continous variables that are either observed or unobserved is very natural to express with Bayesian networks. They provide efficient and formal methods for expressing the joint probability distribution of the model and, moreover, for estimating the parameters and calculating the statistics.

6.3.2 Software for Bayesian Analysis

There are multiple commercial and open source software and toolboxes to carry out Bayesian statistical analysis. In this thesis, two different software tools are used, which are the Bayes Net toolbox (BNET) for MATLAB and the OpenBUGS software. BNET is the most extensive and developed toolbox for Bayesian networks in MATLAB. It is an open source package, but it requires the licence of MATLAB (Murphy, 2001). OpenBUGS is an open source software for Bayesian analysis and can be used to make analysis with many kinds of Bayesian models including the hierarchical models and Bayesian networks (Thomas et al., 2006).

The principal difference between these software tools is that the BNET performs exact analysis using the analytical distributions, whereas the OpenBUGS uses Gibbs Sampling to obtain simulated estimates of the desired statistics. This makes the BNET very restricted in terms of the available distributions. Basically, it can do calculations with only discrete distributions and a few continuous distributions such as Gaussian distribution. This also makes the BNET computationally more inefficient. The OpenBUGS can do analysis with multiple different distributions including continuous, discrete and multivariable. It is also much more efficient computationally and can do calculations much faster. However, using it requires that the user checks the convergence of the Markov chains that are used in the sampling, which makes it more complicated for the user than the BNET.

Chapter 7

Applying Models to Data

Previous chapters have provided the basic background knowledge of the data and the properties of the indices. Moreover, several methods and models were represented that would be useful in constructing an extensive condition monitoring scheme for the harvester head index data. Next the models and methods are applied to the data. This chapter begins with an overview to the distributions of the indices. Knowledge about the distributions helps to decide which data pretreatment methods, discussed in Chapter 4, should be used. In the next step, a deeper look into the structure of the index data is taken. The main characteristics and anomalies are explored and reasons behind them are considered. Careful examination of the data finally produces a sufficient amount of knowledge for constructiong the models described in Chapter 6. After a set of reasonable models are constructed the model diagnostics take place. The analysis in this section differs from the analysis made by Repo et al. (2006) in two manners that are related to the quality of data and the applied methods. The data related factors are that there are no artificial faults cases, the used indices are first level indices and the amount of data in this study is significantly greater. The difference in the used methods will become apparent later, briefly the T^2-statistic and k-means clustering used by Repo et al. (2006) are extended and the methods introduced in preceding chapters are used.

Table 7.1: Names and the categories of the used indices, the abbreviations will be used mostly in the coming figures

#	index name	abbreviation	index category
1	Start Stuck Percentage	StartStuck	Feed acceleration
2	Acceleration Stuck Percentage	AccStuck	
3	Acceleration Delay	AccDelay	
4	Acceleration Time	AccTime	
5	Average Automatic Feeding Speed	AutoFeedSpeed	Feeding speed
6	Automatic Feed Stuck Percentage	FeedStuck	
7	Automatic Feed Approach Time	FeedAppTime	
8	Automatic Feed Approach Length	FeedAppLen	
9	Bucking Success	BuckSucc	Bucking
10	Search Time	SearchTime	
11	Positive Positioning Error	PosError	
12	Negative Positioning Error	NegError	
13	Sawing Index	Saw	Sawing

7.1 Overview on the Index Data Distribution

There are altogether thirteen indices that can be used in the harvester condition monitoring. A list of the indices is shown in Table 7.1. The indices can be divided into three categories based on which phase of operation they are involved in. The phases correspond loosely to the four procedures defined in Section 3.1. In Table 7.1, the categories are named as *feed acceleration*, *feeding speed*, *bucking* and *sawing*. This categorization already represents the initial assumptions about the relations between the indices, however, it is not the only arguable categorization for them. For example, the indices that measure the stuck percentages could be separated into separate category of their own. This would be reasonable especially if the faults connected to stucks are examined. However, the reason for choosing the current categorization will become evident later.

Instead of representing the whole data graphically, a carefully selected sample illustrating the main characteristics of the data is presented. One such sample is plotted in Figures 7.1-7.3, this sample data is named as DATA SET 1 (see Appendix A.1). The scatter plots used here are very illustrative in representating the distribution of multivariate data. They allow pairwise comparison of the indices for detecting correlations and clusters in the data, which increases the understanding of the dependencies in the

process. The data contained in Figures 7.1-7.3 originate from two operators operating on the same harvester. Data points of each operator are plotted with own colors. The corresponding time series data are shown in Appendix A.2. One should pay attention on the time interval between the observations in the time series data. The time between observations is not constant. However, one observation corresponds to approximately 3 hours of work. At first glance, the plots show that some of the indices are saturated, which can be seen especially in the histograms. Particularly, the histograms express high peaks and/or no tails near the saturation point. As seen in Figures 7.1-7.2, in particular, the stuck percentage and the positioning error indices express the saturated values of obsevations. The reason for saturation is that the stuck percentage indices get a value of 100 if no stucks occur and the positioning error indices get a value of 100 if no positioning error occurs. The models that will be used later require distributions that are not limited at one end, discontinuous or not smooth. Therefore, the stuck and positioning error indices are discarded from this analysis. This does not mean that they are not useful for fault diagnosis purposes, they are only inproper for the fault diagnosis models used in this thesis. Excluding the five indices leaves us with eight indices for further analysis. The distributions of these indices seems appropriate for the used models, which require smooth and continuous distributions.

Further examination of the remaining index plots shows that some of the indices are correlated, for example, the feed approach time and length indices as well as the the acceleration time and delay indices. This is expected because these indices measure more or less the same phase of the operation. Similarly, the variation of indices can be different between the operators. This is seen, for example, in the AverageAutomaticFeedingSpeed -index. Another significant feature expressed by the data is that the data points of the different operators seem to be centered at different positions. This is expected as the operators are of different skill levels and, therefore, use different parameter settings. This structure in the data can be thought of as different *operating points* among the operators. Each operator tends to use certain parameter settings and these settings depend mainly on the skill level of the operator and the practices the operator has

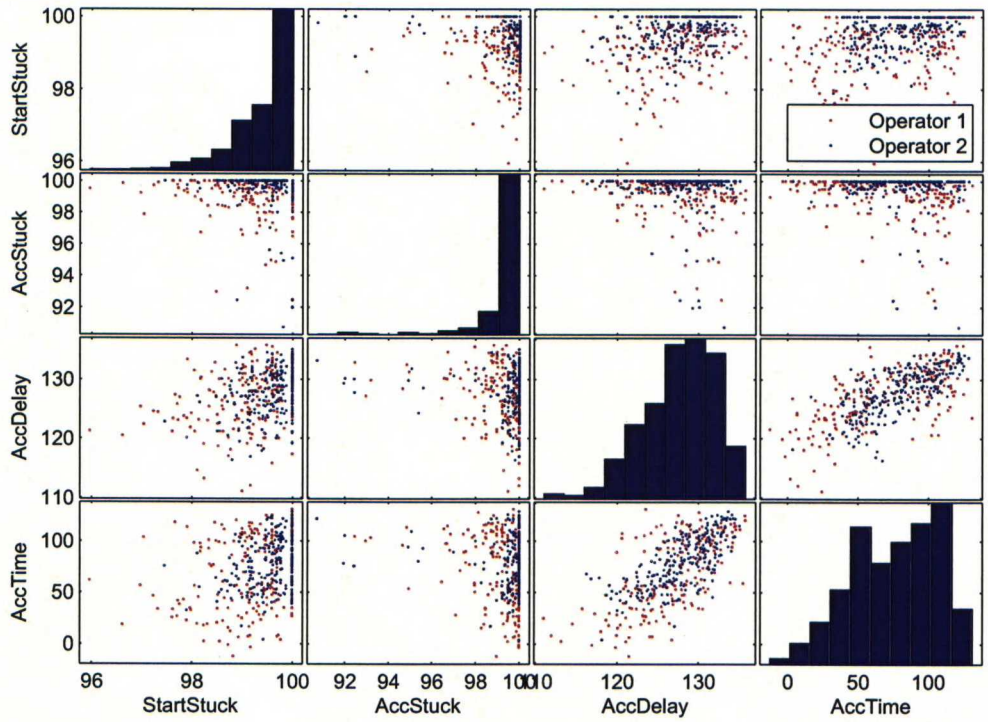


Figure 7.1: Indices related to harveter feed acceleration from DATA SET 1.

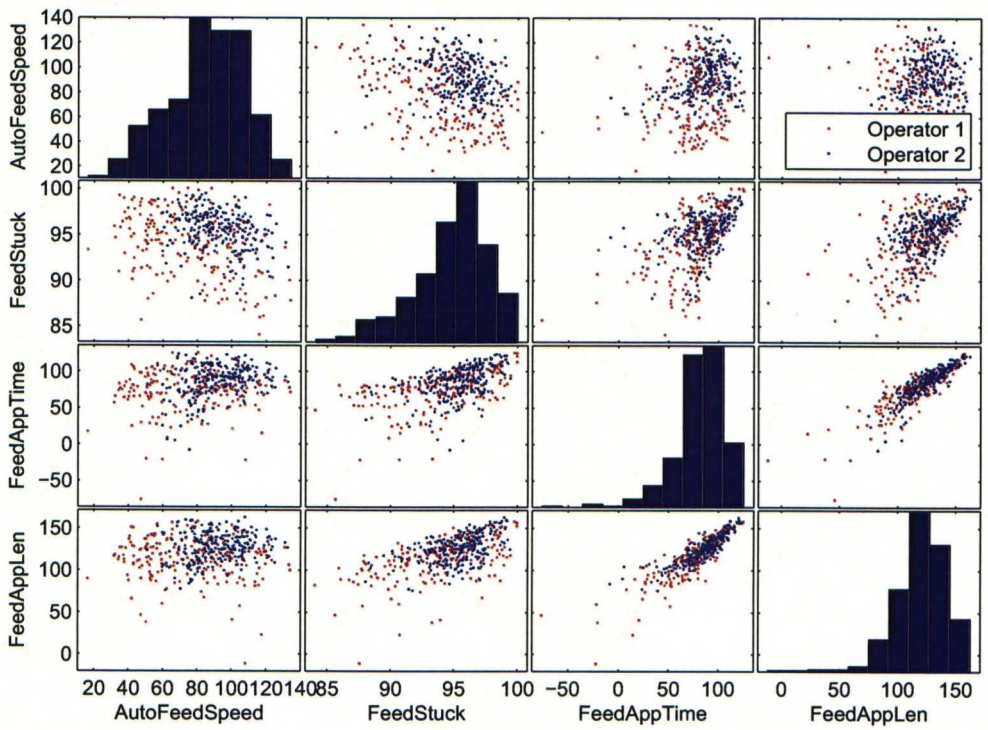


Figure 7.2: Indices related to feeding speed and success from DATA SET 1.

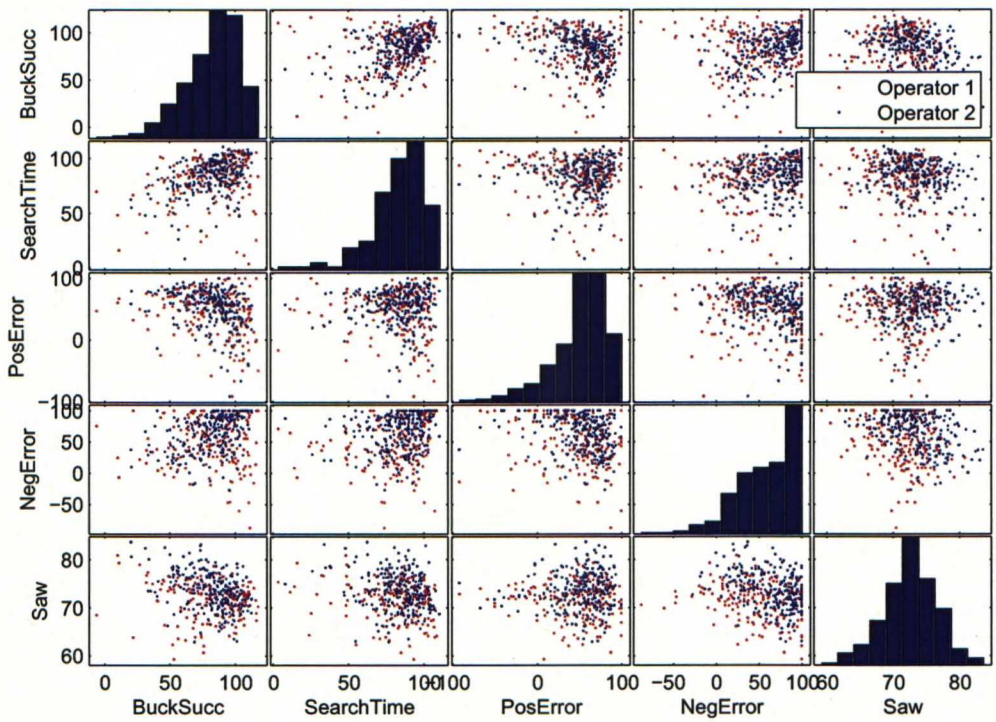


Figure 7.3: Indices related to bucking and sawing from DATA SET 1.

adopted. These operating points are comparable to the in-control states discussed in Section 2.1. However, the operating points of different operators must not be confused with the out-of-control states which occur simultaneously in the data, which can cause difficulties if the existence of the operating points are not considered properly. The prior knowledge provided by the data experts shows that the AccelerationDelay index is very sensitive to errors in the measurement equipment and, therefore, should not be used in data analysis. On the other hand, the AccelerationTime -index measures the performance of the acceleration in a very similar manner, which can be seen as the correlation of these indices. For this reason, one more index, the Acceleration Delay is dropped from the analysis.

The figures shown above do not express clearly the clustered structure in the data of a single operator. Another sample data (DATA SET 2) which shows the clustering more explicitly are shown in Appendix A.3. These data are from four operators operating on the same harvester. The different operating points of each operator seem to be

separated clearly. Taking a more accurate look at the data of an individual operator shows that the data of the operator appear to be clustered also. This is especially noted in the SawingIndex plots. So, there is a reason to make the assumption that the data of each operator have also distributed into operating points, depending on, for instance, the parameter settings of the operator. The observed clustering structure strengthens the initial assumption that the operating points are separated and each of them are observed as an condensation of data points in the observation space $\mathbb{X}$.

At this point it should be pointed out that the operating points between the operators and within the operators are caused mainly by the same reasons, the parameter settings used by the operator. Thus, the analysis of the between and within operating points of the operators will not be distinguished later in this analysis unless mentioned otherwise. Even though the focus will be on the within operator operating points, the analysis methods, however, are the same for both cases.

An overview of the index data distributions provides the following insight into the distribution of the data:

- Observations of each operator are centered at different locations in the observation space $\mathbb{X}$.
- Observations of an individual operator are distributed into operating points in the observation space $\mathbb{X}$.
- Correlation between variables occurs more likely between indices in the same index category than between the indices in different categories.

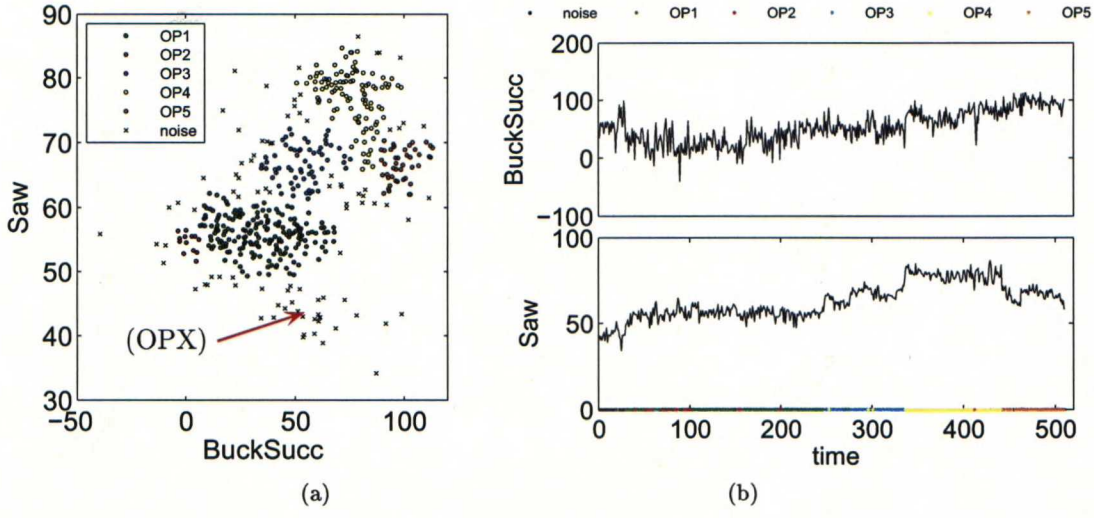
7.2 Index Data Clustering

The clustered structure of the index data gives rise to use the clustering methods introduced in Chapter 5. The clustering methods are used to discover the latent states, i.e. , the operating points, from the data. Particularly, this means that the discovered

clusters are classified into categories that are assigned to either the in-control or out-of-control states, which supports the fault detection in the data. The objective is to find a clustering structure that enables comparison of the new index data with the old data and make a decision based on the allocation of the new indices.

The application of clustering methods forces making a few simplifications in the data. It is not that the clustering methods would require this, but the interpretation of the obtained clustering configurations does not make sense unless these assumptions are made. Firstly, the observations are assumed to be non-autocorrelated, which means that the subsequent observations in each cluster are assumed to be independent of each other. Secondly, the observations in each cluster are assumed to be approximately normally distributed. This is not a very strict assumption and principally it is sufficient that the observations in each cluster are unimodal and approximately symmetrically distributed.

Next the DBSCAN method is applied to the index data represented in Appendix A.3. To keep the presentation simple, only the data of two indices and observations of one operator are used. Figure A.3 shows that the data is most fragmented in the scatter plot of the Sawing and the Bucking Success indices of the Operator 2. These indices are a good starting point for demonstration of the applicability of the DBSCAN algorithm. The parameters ε and N of the DBSCAN were determined using the likelihood ratio method described in Section 6.1.3. Figure 7.4 shows the results of the clustering algorithm. The algorithm classifies the observations into six classes, namely the operating points OP1-OP5 and noise. The OP1-OP5 are the actual clusters and the noise contains the unclassified observations. As seen in Figure 7.4(b), the clustering method distinguishes the operating points of the operator quite well. However, there is evidently an operating point below the OP1 that seems to be classified as noise. Moreover, the OP5 seems to be classified as its own cluster although it could be as well a part of the OP1. Also, some observations of the OP4 seem to be in the area of either the OP3 or OP5. Despite these few faulty classifications, the results strengthen the initial assumption about the operating points. The observations in each cluster are sequential observa-



(a) (b)

Figure 7.4: (a) DBSCAN clustering and (b) the time series of the data of the Operator 2 in DATA SET 2. OP1-OP5 indicates the operating points, OPX is an unclassified cluster.

tions in the time series, as seen by comparing the colored labeling of the observations in Figures 7.4 (a) and (b). This is explained by the fact that the harvester operator usually stays at one operating point for a certain period of time (in-control state) or that some of the faults lasts longer until noticed (out-of-control state). The former occurs because the operator does not change the parameter settings very often and the latter occurs if the faults are almost unnoticeable to the operator and thus cannot be detected without a thorough investigation of the system. An overall interpretation of the states in this case is that the operator has improved his/her performance in sawing and bucking and moved gradually from OP1/OP2 to OP3, OP4 and finally to OP5. Thus, all of the operating points more likely express the different in-control states rather than faults. Despite this, the unclassified cluster denoted by OPX in Figure 7.4 (a) that seems to be more likely an out-of-control state. Generally, it is very apparent that there are less observations recorded from the out-of-control states, because usually a faulty system is repaired quite soon after the fault is detected. Therefore, clusters with only a few observations makes the DBSCAN classify them more likely as noise and this makes the noise observations very interesting from the fault detection point of view.

It has to be remembered that each noise observation is an abnormal deviation from

a specific operating point and, thereby, should be connected to the relevant operating point. This connection is not based on the nearness in the observation space but a nearness in the temporal sense. That is the noise observation is connected to the operating point that is closest in the time series plot. To determine the closest operating point the time difference between each noise observation t_j^{noise} and the clustered observations is calculated and further used to find the minimum distance that is

$$\arg\min_{t_i} |t_i - t_j^{noise}|, \tag{7.1}$$

where t_i is the i^{th} element of the timestamp vector $\mathbf{t}$. However, connecting the noise observations io the clusters does not mean that they are accepted as normal observations. They still remain as noise observations related to out-of-control states, but they are now classified as an out-of-control state of a certain operating point. The noise observations connected to clusters are shown in Figure A.4 in Appendix A.3. The time series plot shows that the noise observations are located near the corresponding operating points as expected. Almost all observations in the beginning of the OP1 are noise observations. This is explained by the unclassified operating point OPX that was discussed earlier. The bottom figure shows that most of the noise observations are classified into the nearest cluster as expected. Nevertheless, there are a few exceptions, especially in the region between the OP3 and OP5.

The DBSCAN algorithm succeeds in separating the operating points in the index data quite well but some improvements are still required to avoid the misclassifications mentioned above. The problems are mainly related to two general problems of the DBSCAN algorithm. The first is the constant density parameter received by the algorithm and the second is that the algorithm is unable to distinguish clusters that are partially overlapping. A fixed density parameter enables the algorithm to find only clusters with a certain minimum density, i.e. , the minimum requirement for finding a cluster is that there is at least one observation in the cluster that has at least N points within the ε-neighborhood. This causes, for example, the algorithm to classify the observations in

OPX as noise although they evidently form a cluster. The second problem, however, the incapability of the algorithm to distinguish partially overlapping clusters, is more severe. It is highly probable that the clusters overlap and managing this requires too much adjusting of the algorithm parameter. Moreover, it is higly probable that the overlapping clusters are of different density. For the above-mentioned reasons the DBSCAN algorithm is not adequete for clustering the index data for fault detection purposes. This is especially the case with higher dimensional data where the clustering cannot be examined visually, making it difficult to analyze the correctness of the clustering.

7.2.1 Discussion

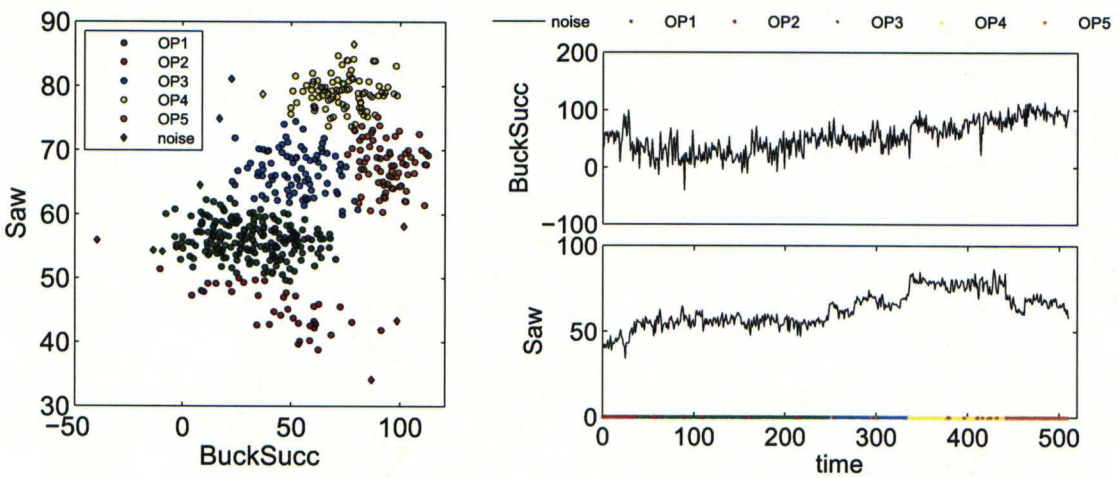
Generally the DBSCAN is very sensitive to parameter variations. Even small adjustments of the clustering parameters causes the configuration of the resulting clusters to vary. From the perspective of the fault detection, it is important that the used method is first of all robust and insensitive to fault alarms. This problem can be partly avoided by the OPTICS algorithm (Ordering Points to Identify the Clustering Structure) (Ankerst et al., 1999). It is an advanced variation of the density based clustering. It is able to handle the clusters of varying density, but it is also unable to distinguish overlapping clusters. Because both of these problems have to be overcome, applying the OPTICS algorithm to data is omitted here and more sophisticated methods are only considered.

7.3 Gaussian Mixture Model

The Gaussian Mixture Model introduced in Chapter 6.2 is a logical extension to the clustering model presented in last section. It is capable of overcoming a few severe problems of the density based clustering, namely the incapability to distinguish clusters with varying density and small mutual distances. However, the results of the DBSCAN algorithm are very useful in the GMM method as prior information about the index data, for example, the amount and locations of the clusters can be predicted from the

DBSCAN clustering results.

Next the Gaussian Mixture Model is fitted into the data. The GMM model parameters are estimated with the EM algorithm, which requires the number of clusters and the initial positions of the clusters as an input. As mentioned, they are obtained from the results provided by the DBSCAN algorithm: the number of clusters equal to the number of clusters given by the DBSCAN and the starting points of the gaussian centers are given by the sample means of the clusters given by the DBSCAN. Each of the initial covariances are set to a unit matrix. First the GMM model is fitted to the data of the operator 2 in DATA SET 2 and the initialization parameters are obtained from the results of the DBSCAN algorithm in last section. In addition, the outliers have been separated by the Hotelling's T^2 statistic described in Section 5.1. The results of the GMM clustering are shown in Figure 7.5. The GMM clustering results seem very



(a) (b)

Figure 7.5: (a) GMM clustering and (b) the time series of the data of Operator 2 in DATA SET 2. OP1-OP5 indicates the operating points.

similar to the results of the DBSCAN clustering. The first remarkable difference is that the cluster OPX in Figure 7.4 is classified as a separete cluster. Actually, the cluster OP2 in Figure 7.4 has been moved to the location of OPX and the former observations of the OP2 have been added to the OP1. This is to be expected from the GMM model because it is quite probable that there should be an operating point at the location of

OPX and it tends to position one cluster there. Another difference in the results of the GMM clustering is that it does not contain as many noise observations as the DBSCAN clustering. This is, of course, dependent on the threshold value T_α^2 of the T^2 statistic which is determined by the confidence level α.

The fairly good and explanatory results obtained by the GMM model with two dimensional data encourages the application of a GMM model with index data with more than two dimensions. Next the GMM model is fitted to four dimensional data consisting of the AccelerationTime, AverageAutomaticFeedingSpeed, BuckingSuccess and Sawing indices of the DATA SET 2. Figure A.5 in Appendix A.4 presents the resulting clusters. The figure shows each pairwise projection of the indices as two dimensional plots and the time series of each index. As seen, the data is divided into five operating points (OP1-OP5) that are partly overlapping in the figure. However, the comparison of different projections indicates that the operating points are located in different parts of the observation space. Furthermore, the figure shows the outliers separated by the T^2 statistic. What can be said on the basis of Figure A.5 is that the operating points seem to be Gaussian although no normality tests were performed for the clusters. In addition, the colored indicator bars at the bottom of the timeseries plots on the diagonal of the Figure A.5 shows that the observations in each cluster are a little more mixed with each other than in the previous clusterings. In other words, the observations in the clusters are no more ordered with respect to their occurence in time but they are assigned to different clusters as if all the observations were mixed totally randomly. One of the main reasons for this is that the operating points are overlapping, which is a weakness of the used methods. This causes the observations at the edges of the clusters to become more likely mixed with the observations in other clusters.

7.3.1 Discussion

The GMM clustering seems to give better results than the DBSCAN clustering in the case of the data in DATA SET 2. The advantage of the GMM clustering compared

to DBSCAN is especially the ability to find operating points with sparse observations. The ability to vary the threshold level for the outlier detection also makes the GMM model more flexible.

It was suspected that the observations would be mixed with each other and the occurence of the observations in the cluster would not be coherent with the temporal order of the observations. This is the most severe drawback of the clustering models used in this and the previous section and makes the models inadequate to be used alone. On the other hand, it is probable that the clustering methods used above will provide useful tools for analysing the results locally, meaning that they provide useful analysis methods in situations where, loosely speaking, the harvester is operating in a stationary state, i.e. , the parameter settings and the surrounding conditions remain unchanged.

Identification of the stationary states in the harvester operation requires the modeling of the dynamical behavior of the harvesting process. This means that the autocorrelation properties and information of the history of the process, i.e. , the changes in parameter settings and the environmental conditions, are modeled. The clustering models used so far in this thesis lack the ability to use history or any temporal information.

7.4 Bayesian Network Model

The Bayesian network model is used to resolve the problems with the clustering models used in the preceding sections, i.e. , the overlapping of clusters and the complexity in combining data from different operators and harvesters. Inspection of the index time series in the preceding sections showed that the clustered structure of the data has also a certain time behavior. This is illustrated clearly in Figure 7.6 which presents the index data of Operator 1 from DATA SET 2. The observations that are subsequent in time tend to be more close to each other also in the observation space, i.e. , there is less variation within a cluster than between all clusters. In Figure 7.6, each set of subsequent observations that are indicated with the same color belong to the same

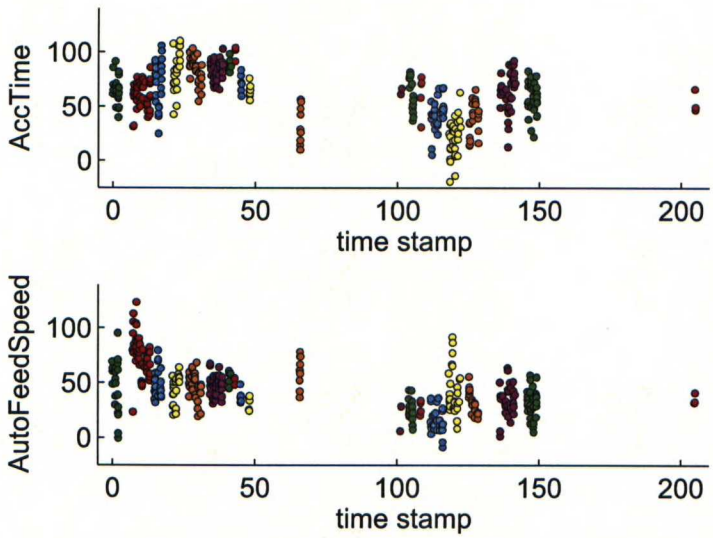


Figure 7.6: Time series of two indices shows the clustered nature of data in time. Each colored set of subsequent observations are observations with constant conditions.

period of operation where the difference between the timestamp values is less than 24 hours. Therefore, each colored cluster represents a period of operation that has occured most likely in the same stand with roughly the same surrounding conditions and using the same parameter settings. Hence, the concentration of the observations within each colored cluster suggests that the conditions within that cluster are static and the observations are generated by a stochastic process with constant parameters. The figure also shows indirectly why the clustering failed in the preceding sections. If the clustering shown here is assumed to be the one that was searched for it is obviously seen that the overlapping of clusters makes it impossible to distinguish them with the methods used.

After discovering the clustering structure described above it is interesting to consider a suitable model and parameters for the stochastic process that generates the observations. Figure 7.6 suggests that the parameters vary significantly between the clusters. Therefore, the parameters can equally be thought of as a realization from another stochastic process, where they are either independent or dependent of each other. Also, for the sake of simplicity it is convenient to consider the parameters and observations to have a certain causal relationship. That is, the observations are assumed to be dependent on

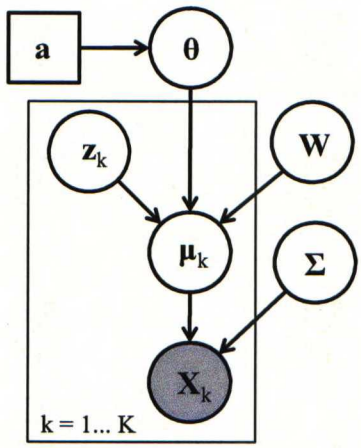


Figure 7.7: The DAG of the hierarchical Bayesian model of the harvester head index data.

parameters and these parameters might be again dependent on some hyper parameters. This kind of hierarchical dependence of parameters is similar to the dependence that is often modeled by the Hierarchical Bayesian models and the Bayesian networks.

The main principles of applying a Bayesian network to fault detection were discussed in Section 6.3.1. In the following, a model containing discrete and continuous variables is presented. The model structure has been adopted from (Bishop, 2006; Gelman et al., 2003; O'Hagan and Forster, 1999; Neapolitan, 2004) and this can be very easily justified based on the index data and the expert knowledge about forest harvesters that has been available. The DAG of the Bayesian network model is represented in Figure 7.7. Only one of the nodes contains the observed variables and the other variables are unobserved. $\mathbf{X}_k$ represents the observations of the index variables X. Particularly, the set $\{\mathbf{X}_k\}_{k=1}^{K}$ is a partition of the data matrix $\mathbf{X}$. The partitioning is based on the observed classification variables A and timestamps $\mathbf{t}$. So, the observations in each group k belong to the same class which means that they are observations from the same stand and operator. The distribution of the observations in each group is assumed to be multinormal, i.e.

$$[\mathbf{X}_k]_i \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \tag{7.2}$$

where $[\mathbf{X}_k]_i$ means the i^{th} row of the observation matrix. Assuming that the obser-

vations are i.i.d. , the joint conditional probability density of the observations is given by

$$p(\mathbf{X}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \propto \prod_{i=1}^{I_k} p([\mathbf{X}_k]_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \tag{7.3}$$

where I_k is the number of observation in the k^{th} group. The unobserved variables are the expected value of each group $\boldsymbol{\mu}_k$, the independent latent variables $\mathbf{z}_k$, the expected value of the cluster centers $\boldsymbol{\theta}$, the discrete variable $\mathbf{a}$ for discrete condition changes and the parameter matrices $\mathbf{W}$ and $\boldsymbol{\Sigma}$. The conditional distributions of the unobserved $\boldsymbol{\mu}_k$s are given by

$$\boldsymbol{\mu}_k|\boldsymbol{\theta}, \mathbf{W}, \mathbf{z}_k \sim N(\boldsymbol{\theta} + \mathbf{W}\mathbf{z}_k, \sigma^2\mathbf{I}) \tag{7.4}$$

and the prior distributions of the unobserved variables are

$$\mathbf{z}_k \sim N(\mathbf{0}, \mathbf{I}) \tag{7.5}$$

$$\boldsymbol{\theta} \sim N(\mathbf{m}, \boldsymbol{\Sigma}_m) \tag{7.6}$$

$$\boldsymbol{\Sigma}^{-1} \sim W(\boldsymbol{\Sigma}_0, l) \tag{7.7}$$

$$[\mathbf{W}]_{.j} \sim N(\mathbf{0}, \mathbf{I}), \tag{7.8}$$

where $\mathbf{m}$ and $\boldsymbol{\Sigma}_\mathbf{m}$ are constant parameters. The model is not very self-explanatory, so a short description of each part of it is provided. The expected value $\boldsymbol{\theta}$ of the cluster centers indicates the mean value of the data from all operators having different continuous parameter settings and environmental effects. The discrete parameter vector $\mathbf{a}$ has M elements, each obtaining a value of 1 with the probability π_m, i.e. , $p(a_m = 1) = \pi_m$ for each $m = 1 \ldots M$, such that $\sum_{m=1}^{M} \pi_m = 1$. Each group has its own group mean $\boldsymbol{\mu}_k$ but the covariance matrix $\boldsymbol{\Sigma}$ is common to each group. Each group also contains the latent vector $\mathbf{z}_k$ which is used to obtain the mean of the unobserved group mean $\boldsymbol{\mu}_k$. The hidden discrete state variable $\mathbf{a}$ is omitted in this case for simplicity, i.e. , it will have only one value with a probability of one. The analogy of the hidden variables $\mathbf{z}_k$ to the principal components $\mathbf{z}$ presented in Section 5.2 is important to notice. Both of these variables are the uncorrelated hidden variables of the observed

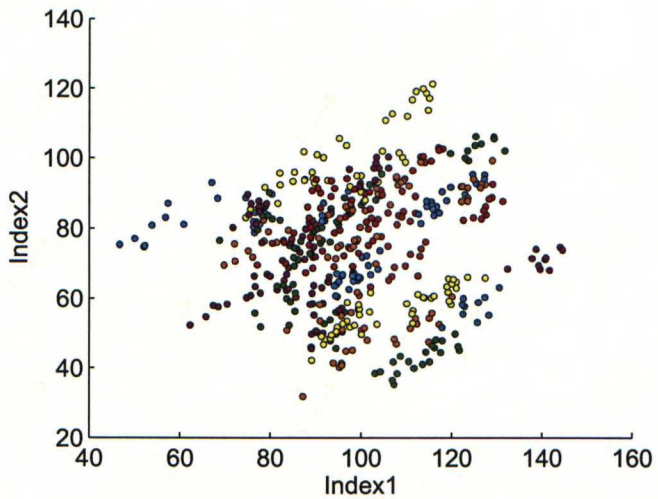


Figure 7.8: Simulated data

variables. The Bayesian network model presented here is also called a Bayesian PCA model, which explains this analogy. An important part of Bayesian analysis is choosing the prior distributions of parameters. Probably, the safest way is to choose the generally known *conjugate prior distribution* as a prior distribution for each parameter. In the case that the posterior distributions are estimated with simulation this requirement is not equally important as in the case that the posterior distributions are determined symbolically. However, using the generally known conjugate prior guarantees that the prior distributions are proper, i.e. , the posterior distribution integrates up to a finite number (Gelman et al., 2003).

A good starting point for testing the above model, is to make first a simple model involving only a few variables. Particularly, this means that the number of observed variables is one or two and the number of unobserved variables is one. In the following the model is estimated using simulated data. The data contains observations of two variables that are correlated and the data is divided into $K = 50$ groups. The number of hidden variables in $\mathbf{z}_k$ is one. The motivation for using simulated data is to find the proper prior distributions to minimize the efforts required when the actual data is used. The simulated data replicates the behavior of the two observed variables AccTime and AutoFeedSpeed in the original data. The simulated data is plotted in Figure 7.8. The

parameter values used in the simulations are

$$\boldsymbol{\theta} = \begin{bmatrix} 100 \\ 80 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 13 & 10 \\ 10 & 16 \end{bmatrix} \quad \text{and} \quad \sigma = 16. \tag{7.9}$$

The model described by Figure 7.7 and (7.2) - (7.9) is implemented as an OpenBUGS model. The resulting model description is shown in Appnedix B.1. Simulations with the model produces the posterior density estimates shown in Figure B.1 in the Appendices. The posterior distribution estimates of $\mathbf{W}$ shown in the top figures seem to be rather bimodal and not very close to normally distributed. The mean value $\bar{\mathbf{W}} = [\ 5 \quad 5\ ]^T$ does not seem to be very likely. This is probably caused by either bad formulation of the model or the insufficiency of the given data. Bad formulation of the model is most surely related to the prior distribution (7.8) of parameter $\mathbf{W}$. The given prior distribution allows the vector to have both negative and positive value, i.e. , the models is unable to separate the mean values $\bar{\mathbf{W}} = [\ 5 \quad 5\ ]^T$ and $\bar{\mathbf{W}} = [\ -5 \quad -5\ ]^T$ and they are both considered as possible means for the distribution of $\mathbf{W}$. The second assumption about the insufficiency of the data is not as well argued, but still possible since there are quite many distributions of parameters estimated from only a few observed variables. At this point, it seems impractical to estimate the Bayesian network model with the actual data using the OpenBUGS software because of the above-mentioned unsolved problems.

CHAPTER 8

Summary of Results and Discussion

Several helpful methods for manipulating and analyzing multivariate index data from a forest harvester head were introduced in the theoretical part of this thesis. In the experimental part, three different multivariate statistical methods and models were tested in practise, and their applicability and properties were analyzed. This chapter summarizes the main results and findings of this study and finally some proposals for further studies are given.

8.1 Presumptions and Conclusions

An in-depth discussion with the forest harvester experts preceded the investigation of the harvester head index data. According to the experts, the behavior of the forest harvesting process that is expected to be seen in the index data is, that certain operation points could be seen as distinguishable clusters. These operation points are presumably caused by different internal and external variations of the process that are generally named as the special cause variations (see Section 2.2.1). The internal variations are caused, e.g. , by the skill level of the harvester operator, the used parameter configuration and the technical condition of the harvester. The external variations are caused, e.g. , by the environmental variations such as the weather conditions and the forest base condition. In the first place, it is assumed that the operation points can be separated from each other. This means that the internal and external factors that cause

the data generating process to make distinguishable observations behave in a somewhat discrete manner. This assumption is very reasonable if we consider, for instance, the differences caused by a change of operator which can be thought of as a transition from one state to another in the harvester data generating process. Switching the operator certainly causes discrete changes in most of the internal factors that are operator dependent. This model contains an assumption that the continuous internal and external variations are dominated by the variations caused by the discrete state changes, i.e. , the effects caused by the continuous variables are some order of magnitude smaller than the effects of discrete variations. Particularly, this means that in the observed index data the observations related to different states are distinguishable.

The harvester head index data available in this study were gathered from 140 harvesters that were operated by more than 350 operators. Most of the data were reviewed to find the most suitable observations for model identification purposes. The models were tested only on a small part of the data, but there is no reason that the results would contradict with the rest of the data. The index data contains only qualitative and quantitative information about the harvester head performance. There are no data about the environmental conditions, the operator decisions or the technical condition of the machine. Therefore, the used models are identified by using unsupervised learning, which means that the capable analysis methods are restricted to the data-driven methods. The inferences are made only based on the characteristics found from the data and these results are then compared to the expert information about the harvesting process.

8.2 Results of the Clustering Methods

Concerning the distinguishable operation points in the index data as a consequence of the discrete variations in the harvesting process, it is necessary to be able to extract the operation points from the data in order to make further interpretations and conclusions about the state of the harvester head operation. Suitable methods for finding the operation points from the data are certain statistical clustering techniques or al-

gorithms. However, because there are noise and outlier observations in the data, the most traditional clustering methods, such as hierarchical and partitioning clustering, will fail. Instead, the density based clustering algorithms are capable of overcoming the shortcomings of the traditional methods. A simple density based algorithm, called DBSCAN is introduced and applied to the data. It has been proven to be a considerably more efficient method for clustering data with noise or outliers (Ester et al., 1996). Results show that the DBSCAN algotihm is capable of distinguishing the operation points quite reliably in some particular cases. However, in most cases the algorithm fails to distinguish the operation points, which is mainly because they are overlapping. Although the algorithm itself is insufficient for separating the operation points, the results given by the DBSCAN confirm the assumptions about the existence of the operation points. These results support the decision to make further analysis with more developed methods that concentrate on similar assumptions about the operation points.

8.3 Results of the Mixture Model

The Gaussian mixture model is used to obtain better results in separating the operation points from each other. GMM is a probabilistic model and, therefore, it provides a probability distribution of the random variables in the data generating process by using the observed data. The model parameters can be estimated using the information given by the DBSCAN algorithm or it can be estimated without this information which naturally requires more computational effort in comparing different model candidates. The algorithm, however, requires that the number and expected positions of the clusters are given as initial parameters. The optimal cluster configuration can be thereafter found by comparing the results obtained with different initial parameters.

In this thesis, two GMM models are estimated using the data in DATA SET 2. The number of used index variables is two in the first case and four in the second case. The results demonstrate that the GMM is slightly better in separating clusters than the density based clustering method DBSCAN. The first advantage of the GMM is that it

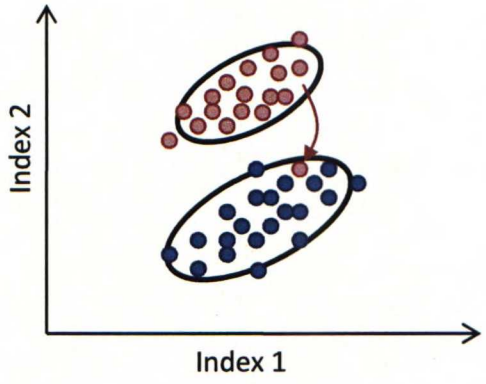


Figure 8.1: Classification problem: one observation of the top cluster, which indicates deterioration in respect of Index2 in the 'red' operation point, becomes wrongly classified to the bottom cluster

is capable of separating clusters with different densities. Therefore, it does not occur so easily that the clusters with only a few observations are classified as noise. Secondly, the GMM allows that the clusters have a more natural form, the normal form. This makes the results of the GMM model less sensitive to the initial parameters than the density based clustering model. However, the GMM does have some disadvantages that makes it also quite inefficient in separating the clusters. Probably the most severe problems are related to the overlapping of the clusters, that seems to be very common in the data. This overlapping can be divided into two types, namely *spatial overlapping* and *temporal overlapping*. Generally, the GMM is incapable of separating clusters that are overlapping in either sense. The spatial overlapping appears as closeness of clusters in the observation space. Whereas the temporal clustering appears when observations that are assumed to be fault state operation from one cluster become mixed with the normal observations of another cluster. Figure 8.1 illustrates an occurence of the temporal overlapping. It is evident that when the clusters overlap noticably in a spatial sense, the risk that normal state observations will be classified to the wrong cluster will increase. Therefore, in the worst case, the two types of overlapping cannot be distinguished from each other.

The results provided by the GMM model motivates the development of methods that are capable of handling the overlapping of the clusters. This will require that the assumption

of entirely discrete state changes in the harvesting process and, consequently, in the model of the data generating process is relaxed. The solution will be that also time continuous state variables are added to the model. This is the key observation that leads to the conclusion that the clustering and mixture models are inadequate to model the harvester head process. Furthermore, this result is the starting point when the Bayesian network models are incorporated to the analysis.

8.4 Results of the Bayesian Network Model

The Bayesian network model is used to overcome the problems related to the GMM as explained in the preceding section. Most of the problems are caused by the overlapping of the operation points. As a solution to this problem, the Bayesian network model introduces continuous hidden state variables that expresses the operation points as continuous states. The discrete states are, however, still retained in the model to describe the discrete state changes, since it is anticipated that such changes still exist. Moreover, a temporal clustering of the observations takes place. The Bayesian network is still unable to resolve the problem with the time dependence of the observations, i.e. , the model does not explain any dynamical behavior in the process.

The basic idea of selecting the form of the used Bayesian network is that the data shown in Figure 7.6 is assumed to be generated by a model that involves at least two levels of parameters. Most of these parameters are continuous but some discrete parameters are also included. Almost all of the parameters, as well as the observations, are assumed to be normally distributed. This is because the normal distribution model is simple and very common. Furthermore, it allows efficient parameter estimation and simulation methods to be used. The prior distributions are chosen to be the conjugate prior distributions, in order to guarantee the desirable properties for the posterior distributions. The model used in this thesis is presented in Figure 7.7.

The posterior distributions of the parameters in the Bayes network model are estimated

using the OpenBUGS software. First, a simulated data is used to prove the capability of the model and the estimation methods. The results given by the simulated data, however, show that some of the presumptions in the model structure are inappropriate. One possible explanation is that the posterior distribution of the parameter matrix $\mathbf{W}$ is invariant to the sign changes of the column vectors of $\mathbf{W}$. This causes the estimated posterior distribution to be strongly bimodal. This problem could be solved, for example, by constraining the distribution of $\mathbf{W}$. Another possible explanation is that the observed data is simply not informative enough to sufficiently explain the total structure of the model.

8.5 Discussion

Examination of the data and applying the introduced models provided valuable insight into the forest harvester head processes. Different models were capable of discovering different characteristics from the data. They also brought out the parts of the data processing that are in need of improvements in order to make more extensive data-driven condition monitoring. Some of the difficulties that were faced in the model parameter estimation of the Bayesian network models remained without clear explanation at the time of finishing the experiments. The results obtained, however, provide important information about the forest harvesting process which can be used as a starting point for further studies of the data-driven condition monitoring of the harvester head. In this final section the conclusions are briefly summarized.

In addition to the results mentioned in previous sections of this chapter, some of the key observations are emphasized here. As noted earlier, different operators at different time instants seem to produce index values with remarkably different characteristics such as mean value, variance and covariation. From the data generating process point of view, this means that some of the parameters are changed. These changes in the process parameters are connected to one or more of the following changes in the harvesting process

- The harvester operator is changed to another
- The operator changes his/her working practises
- The machine parameter settings are changed
- Environmental conditions change

The above changes are thought of as a change of the operation point in the harvesting process. So, each change in the physical process corresponds to a change of one or more of the parameters in the data generating process. It is highly probable that there exists some kind of dependencies between these operation points. For example, under certain environmental conditions the operator always uses a certain practise or certain parameter settings. These dependencies are also seen in the data, e.g. , for some indices the index variance seems to be smaller for operators with a higher mean value of the index and vice versa. This kind of systematic behavior in the dependencies between the indices suggests that some kind of hidden variables, that are dependent on the above mentioned internal and external change, determine the dominating behavior of the observed index values.

Evidently, measuring and monitoring some of the factors listed above can be quite difficult. As discussed in Sections 3.2 and 3.3, the environmental factors and the operator working practises are very difficult to measure and monitor. However, presumably these factors have a remarkably effect on the behavior seen in the index data. Consider, for example, an operator who performs the felling and feeding work phases simultaneously. This kind of practise will definitely improve the feeding index value and make the operator more efficient. According to forest harvester experts, experienced operators tend to do these kind of actions more easily than inexperienced operators. However, if an improvement caused by different kinds of working practices cannot be separated from the improvement or degradation caused by the harvester technical condition, it makes the condition monitoring very insensitive. Practically, this means that the anomalies that can be detected by the condition monitoring methods must be extremely abnormal. Therefore, small deviations that may be more interesting will not be detected at all.

Given index data that were based on carefully selected measurements application of the statistical methods as described in Chapter 4 was found out to be insufficient for using with the tested models. The index data is highly informative and useful for a human decision making, but the complexity of the process and the noise in the data makes it very challenging to apply the data-driven models used in this thesis. It is recommended that the operators' influences on the data, that has already been modeled in the case of forest harvester, are examined from the condition monitoring point of view. With the help of more sophisticated methods developed recently by (Palmroth et al., 2009) and (Tervo et al., 2008), it will be possible to determine the effects of the operator and his/her skills on the process and compensate them more efficiently in the future.

References

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, 1999. ISSN 0163-5808.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006. ISBN 0387310738.

L. H. Chiang, E. L. Russell, and R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2001.

J.-H. Choi, J.-M. Lee, S. W. Choi, D. Lee, and I.-B. Lee. Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 60(1):279 – 288, 2005. ISSN 0009-2509.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL `http://www.jstor.org/stable/2984875`.

K. Detroja, R. Gudi, and S. Patwardhan. A possibilistic clustering approach to novel fault detection and isolation. *Journal of Process Control*, 16(10):1055 – 1073, 2006. ISSN 0959-1524.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering

clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. pages 139–147. Morgan Kaufmann, 1998.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman and Hall/CRC, 2003. ISBN 158488388X.

T. J. Harris, C. T. Seppala, and L. D. Desborough. A review of performance monitoring and assessment techniques for univariate and multivariate control systems. *Journal of Process Control*, 9(1):1 – 17, 1999. ISSN 0959-1524.

V. Hölttä. Analysis of stem diameter measurement in forest harvester. Master's thesis, Helsinki University of Technology, 2004.

V. Hölttä, M. Repo, L. Palmroth, and A. Putkonen. Index-based performance assesment and condition monitoring of a mobile working machine. In *Proceedings of IDETC'05*, 2005.

J. E. Jackson. *A User's Guide to Principal Components.* John Wiley & Sons, Inc., 2003.

H.-P. Kriegel and M. Pfeifle. Hierarchical density-based clustering of uncertain data. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 689–692, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2278-5.

Y.-H. Lee, K. G. Min, C. Han, K. S. Chang, and T. H. Choi. Process improvement methodology based on multivariate statistical analysis methods. *Control Engineering Practice*, 12(8):945 – 961, 2004. ISSN 0967-0661. Special Section on Emerging Technologies for Active Noise and Vibration Control Systems.

U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *In Proc. AAAI*, pages 531–537, 2000.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis.* Academic Press, 1979.

J. Matsuura and T. Yoneyama. Learning bayesian networks for fault detection. *Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pages 133–142, 29 2004-Oct. 1 2004. ISSN 1551-2541.

K. P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33:2001, 2001.

R. E. Neapolitan. *Learning Bayesian Networks.* Prentice Hall, April 2004. ISBN 0130125342.

B. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control.* Oxford University Press, New York, 1994.

A. O'Hagan and J. Forster. *Kendall's Advanced Theory of Statistics, volume 2B, Bayesian Inference, 2nd edn.* A Hodder Arnold Publication, 1999.

H. Ovaskainen. *Timber harvester operators' working technique in first thinning and the importance of cognitive abilities on work productivity.* PhD thesis, University of Joensuu, 2009. URL `http://www.metla.fi/dissertationes/df79.htm`.

L. Palmroth, K. Tervo, and A. Putkonen. Intelligent coaching of mobile working machine operators. In *IEEE 13th International Conference on Intelligent Engineering Systems*, 2009.

A. Raich and A. Çinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE Journal*, 42:995–1009, 1996.

M. Repo. Data-driven condition monitoring of forest harvester heads. Master's thesis, Helsinki University of Technology, 2008.

M. Repo, V. Hölttä, and L. Palmroth. Unsupervised fault detection of forest harvester head functions. In *Preprints of SAFEPROCESS 2006*, Beijing, P.R. China, 2006. IFAC. 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical

Processes - SAFEPROCESS 2006, August 30 - September 1, 2006, Beijing, P.R. China.

S. Sharma. *Applied Multivariate Techniques.* John Wiley & Sons, Inc., 1996.

S. S. Stevens. On the theory of scales of measurement. *Science*, 103:677–680, 1946.

K. Tervo, L. Palmroth, V. Hölttä, and A. Putkonen. Improving operator skills with productivity model feedback. In *Proceedings of the 17th World Congress.* IFAC, 2008.

A. Thomas, B. O'Hara, U. Ligges, and S. Sturtz. Making BUGS open. *R News - The Newsletter of the R project*, 6(1):12–17, March 2006.

Timberjack 1. Timbermatic 300 käyttöohjekirja, versio 1.0. Technical report, Timberjack, 2002.

Timberjack 2. Harvesteripää 725 käyttö- ja huolto-ohjekirja. Technical report, Timberjack, 2002.

K. Väätäinen, H. Ovaskainen, P. Ranta, and A. Ala-Fossi. *Hakkuukonekuljettajan hiljaisen tiedon merkitys hakkuutulokseen työpistetasolla.* Metsäntutkimuslaitoksen tiedonantoja 937. 90 s., 2005.

Appendix A: Additional Figures

A.1 Summary Statistics of the Data Used in the Figures

DATA SET 1:	Number of operators: 2		
	Number of observations	Start time	End time
Total	452	22 May 2006	13 Oct 2006
Operator 1	200	22 May 2006	13 Oct 2006
Operator 2	252	22 May 2006	13 Oct 2006

DATA SET 2:	Number of operators: 4		
	Number of observations	Start time	End time
Total	1182	11 Apr 2006	17 Nov 2006
Operator 1	396	11 Apr 2006	02 Nov 2006
Operator 2	510	11 Apr 2006	12 Nov 2006
Operator 3	245	18 Jul 2006	17 Nov 2006
Operator 4	31	13 Nov 2006	17 Nov 2006

A.2 Time series of the index data: DATA SET 1

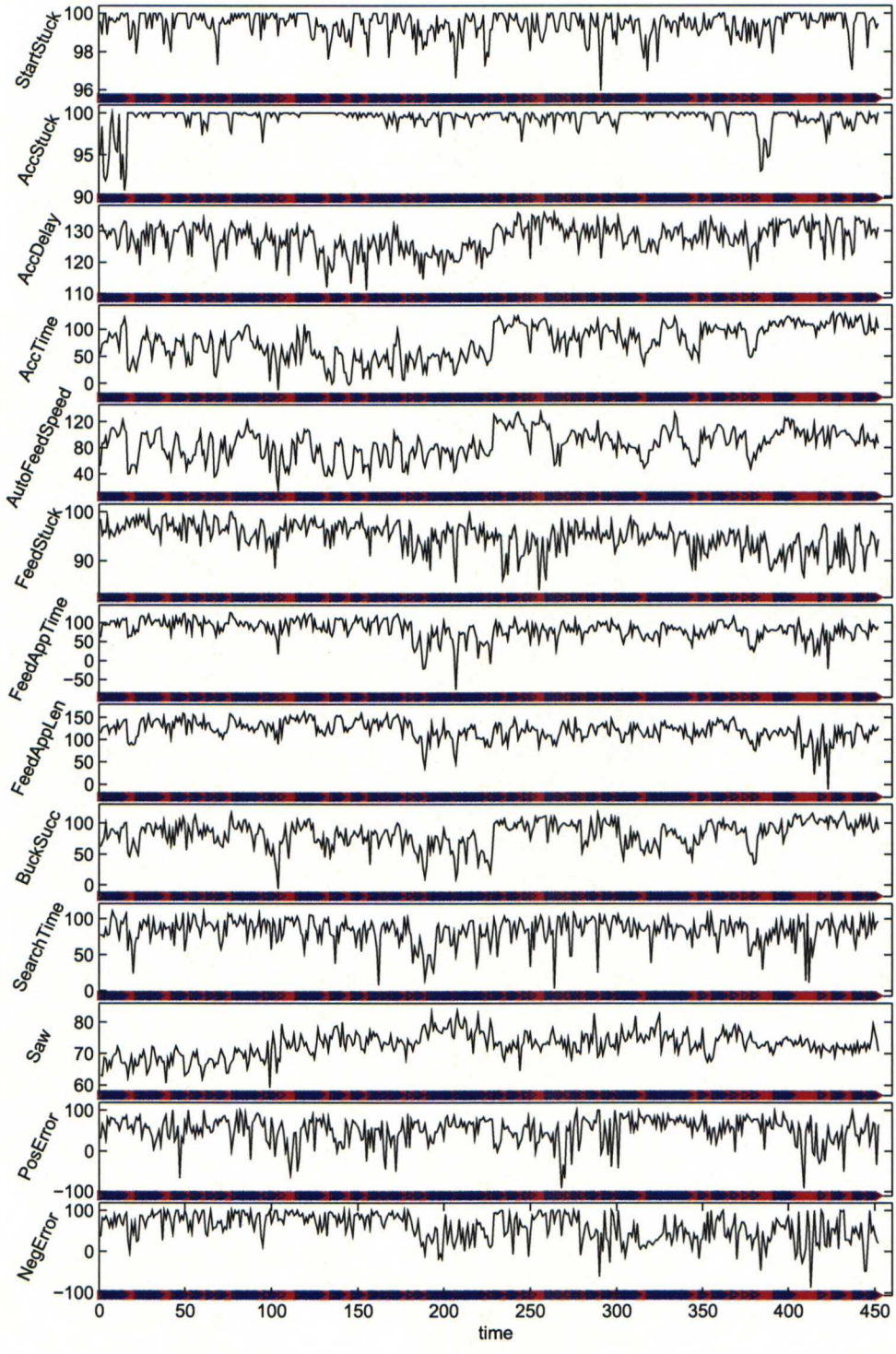


Figure A.2: Time series of the index data shown in Figures 7.1-7.2. Colored bar at the bottom of each plot indicates the operator (red = Operator 1 and blue = Operator 2).The unit of the time is approximately 3 h of work.

A.3 Scatter Plots of Indices: DATA SET 2

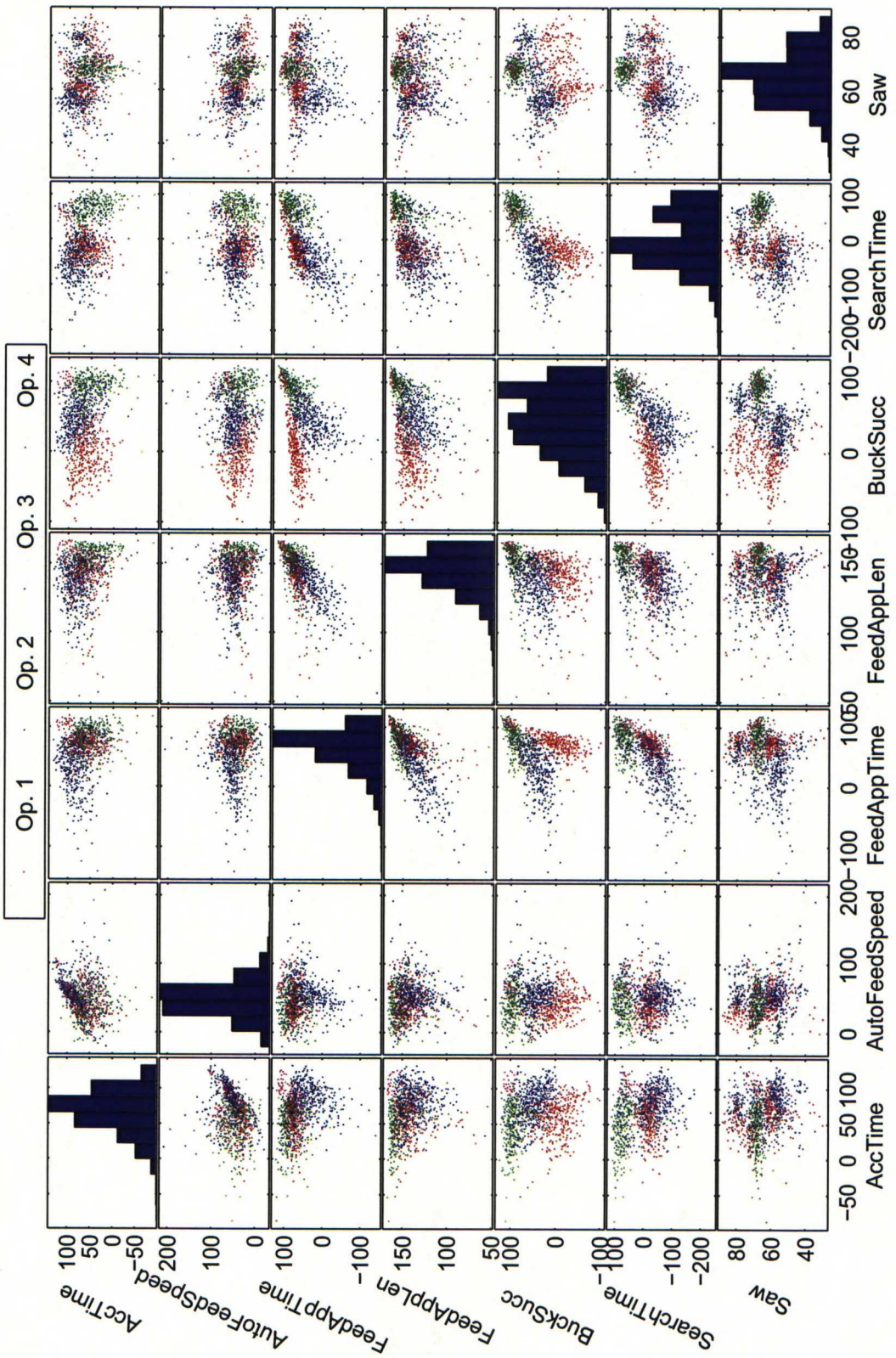
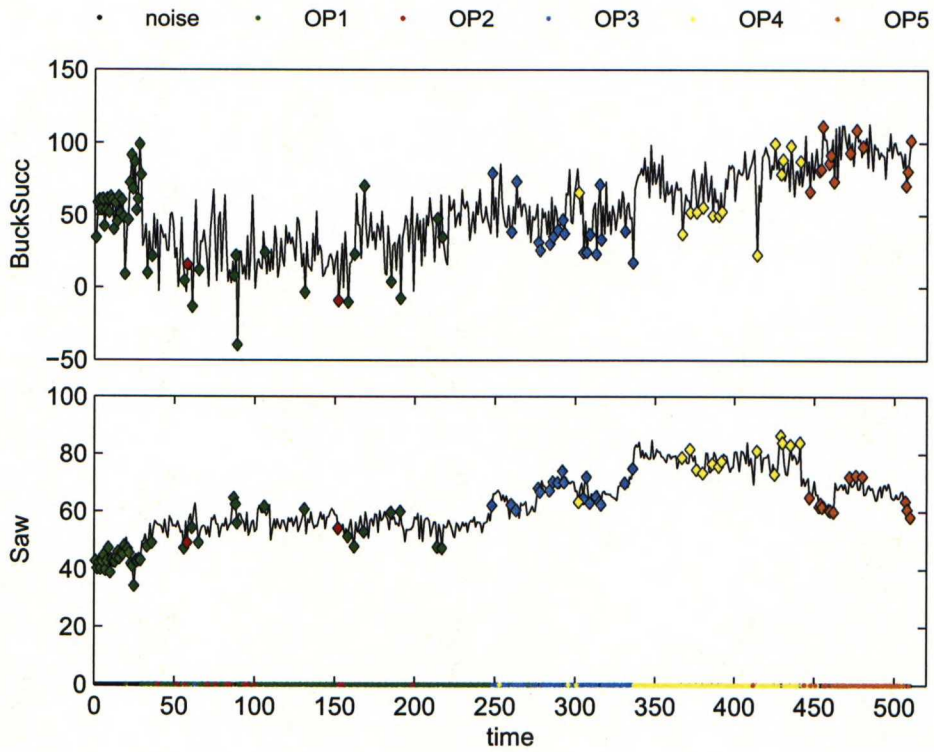
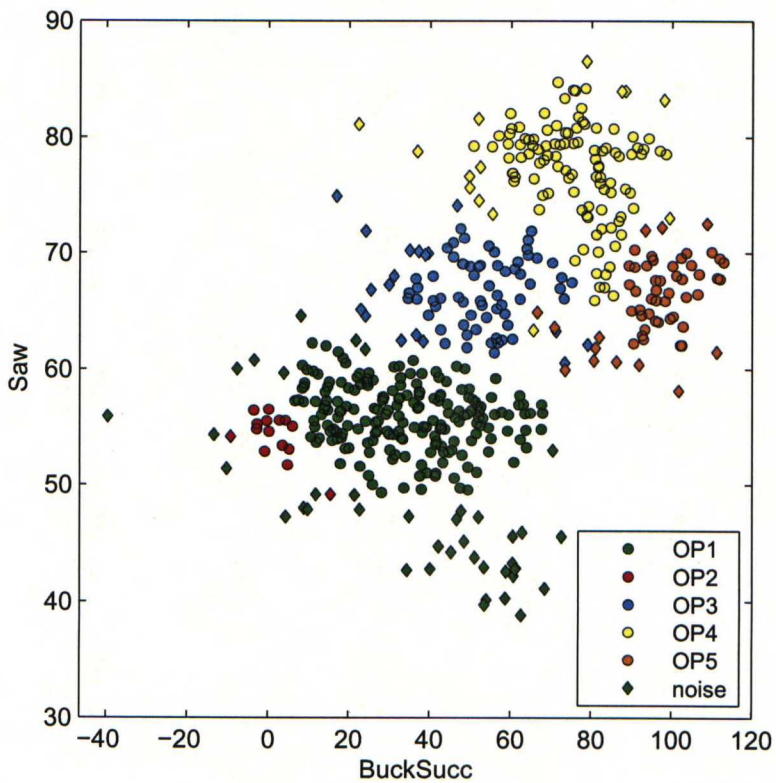


Figure A.3: The index data of four operators on one harvester



(a)



(b)

Figure A.4: (a) The time series and (b) the scatterplot figures with outliers classified into relevant clusters. The diamond shaped (◊) observations are noise.

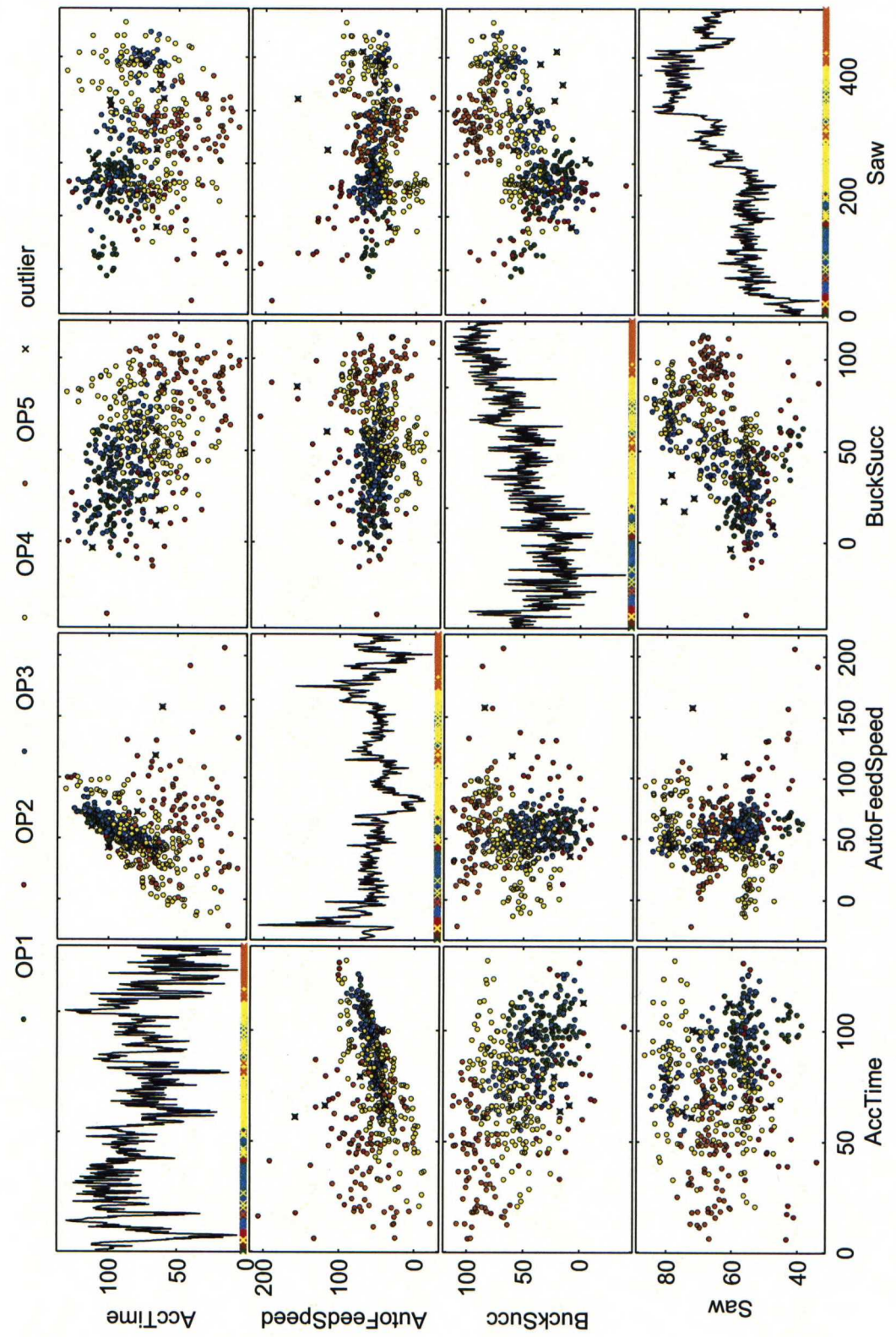


Figure A.5: Results of the GMM clustering for operator 2 in DATA SET 2. Groups with different colors indicate the clusters and × are outliers.

Appendix B: Other Additional Material

B.1 The OpenBUGS model for a Bayesian network

```
model {
    for (i in 1:N) {
        for (j in 1:sizes[i]) {
            y[i, j, 1:2] ~ dmnorm(mu[i,1:2], tauC[1:2,1:2])
        }
        mu[i,1] ~ dnorm(sum[i,1], 0.1)
        mu[i,2] ~ dnorm(sum[i,2], 0.1)
        z[i] ~ dnorm(m, 1)

        sum[i,1] <- theta[1] + z[i]*W[1]
        sum[i,2] <- theta[2] + z[i]*W[2]
    }
    W[1] ~ dnorm(0, 0.1)
    W[2] ~ dnorm(0, 0.1)
    theta[1] ~ dnorm(100, 0.1)
    theta[2] ~ dnorm(80, 0.1)

    tauC[1:2, 1:2] ~ dwish(R[1:2, 1:2], 3)
    m ~dnorm(0,0.1)

    sigmaC[1:2, 1:2] <- inverse(tauC[ , ])
    rhoC <- sigmaC[1, 2]/sqrt(sigmaC[1, 1] * sigmaC[2, 2])
}
```

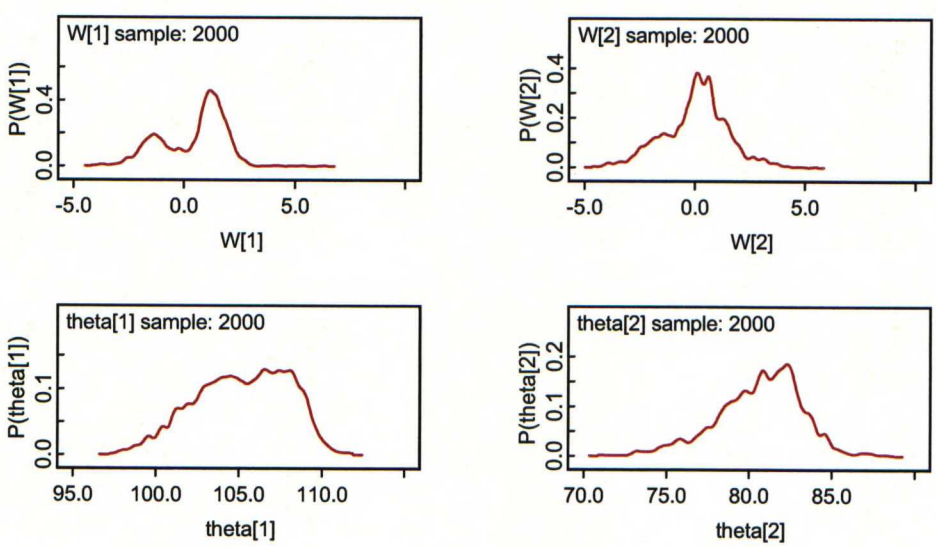


Figure B.1: Estimated posterior distribution of variables $\mathbf{W}$ and $\boldsymbol{\theta}$ when simulated data is used